Analyse, conception et réalisation d'un environnement pour le pilotage et la visualisation en ligne de simulations numériques parallèles

Aurélien Esnard

Université Bordeaux I – LaBRI & INRIA Futurs

12 décembre 2005

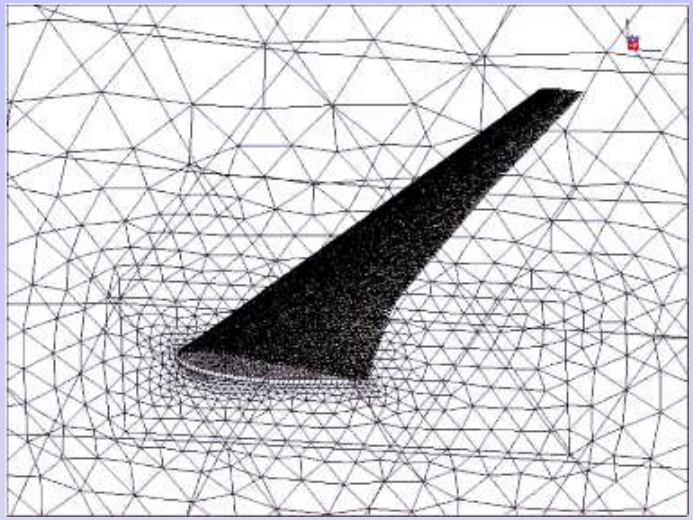
Plan

Plan

Simulation numérique

Programme permettant de reproduire numériquement sur un ordinateur un phénomène physique décrit par un modèle mathématique.

- Phènomènes modélisés complexes
  → mécanique des fluides, biologie moléculaire, astrophysique, ...
  → loi d'évolution en temps (EDP)
- Caractéristiques des simulations
  → très calculatoires (algorithmique complexe)
  → gros volumes de données (maillages fins)
- Programmes parallèles
  → distribuer les données en mémoire
  → plusieurs CPUs pour accélérer les calculs

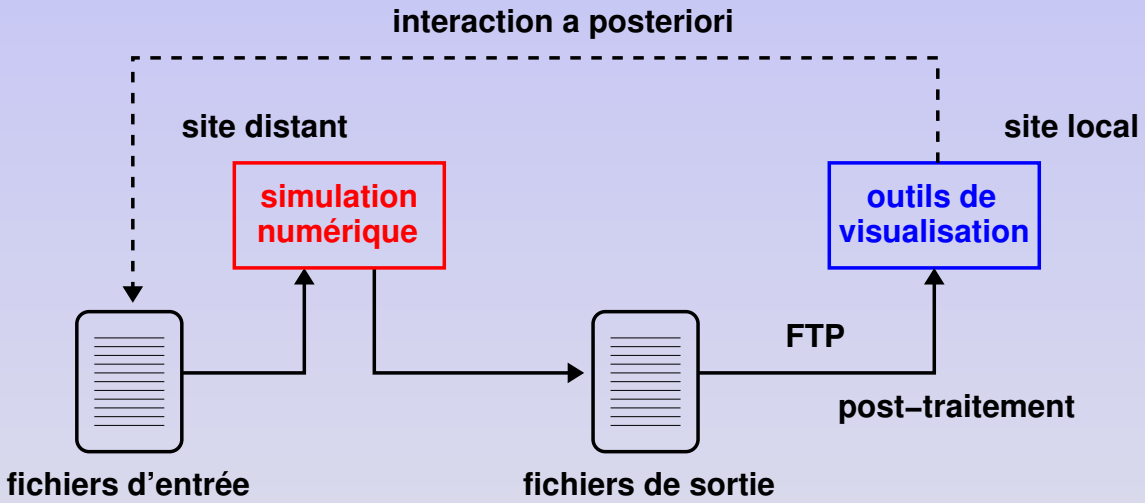


Maillage représentant le profil d'une aile d'avion.

Besoin d'outils pour l'analyse des résultats : visualisation scientifique

Cycle de travail en Batch

- Choix paramètres d'entrée → calcul → analyse des fichiers en sortie
- Modification des paramètres en entrée → calcul → ...

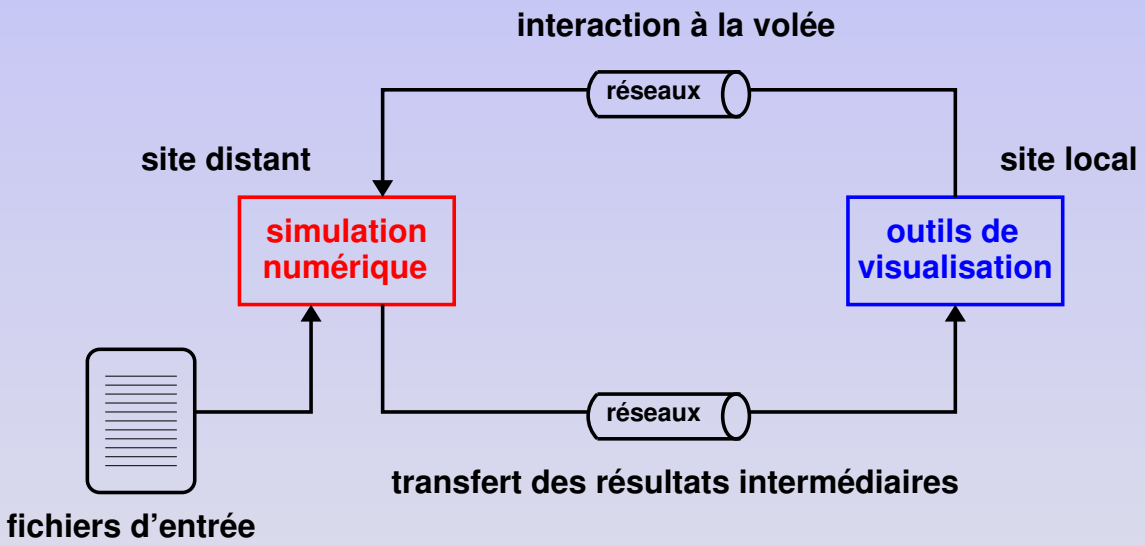


Cycle de travail en Batch long et fastidieux.

Vers le pilotage des simulations

De nouveaux besoins apparaissent... plus d'interactivité !

- Visualiser « en temps-réel » les résultats intermédiaires
- Interagir à distance avec la simulation



Couplage par le réseau de la simulation et de la visualisation

Le pilotage, pourquoi ?

Pourquoi c'est utile ?

- Surveiller l'évolution des calculs dans le cas de simulation longue
  → connexion/déconnexion, détection d'erreurs (ex. paramètres d'entrée erronés)
- Diminuer le temps entre le calcul et l'interprétation
  → visualisation en ligne des résultats intermédiaires
- Retrouver une démarche expérimentale (interaction + retour visuel)
  → explorer l'espace des paramètres, analyse de cause-à-effet

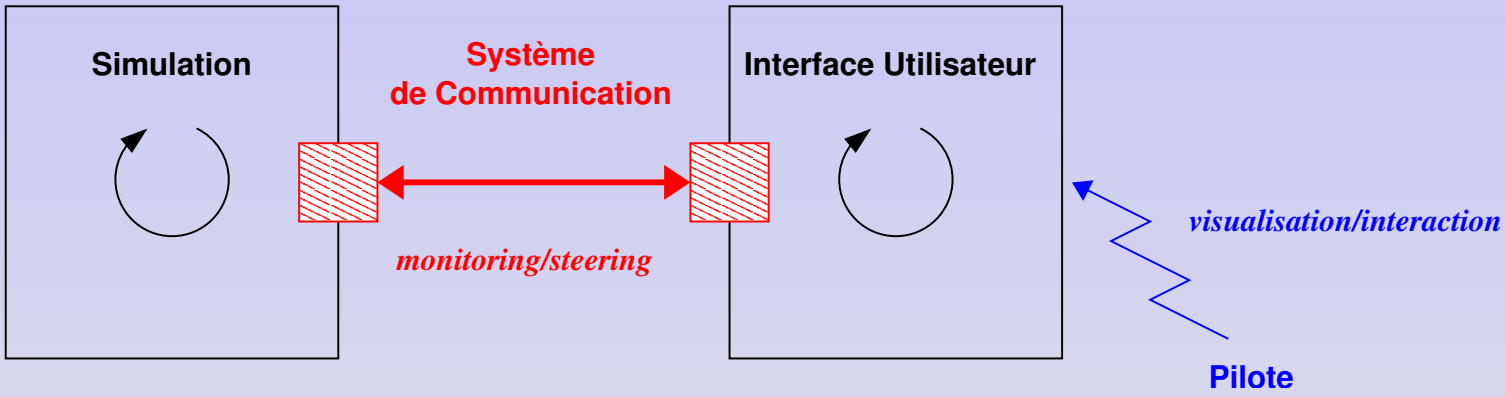
Pourquoi c'est difficile ?

- Coordonner les opérations de pilotage en parallèle
  → garantir la cohérence des opérations de pilotage alors que la simulation n'est pas synchronisée
- Transférer et visualiser rapidement un gros volume de données
  → limiter le surcoût induit par le pilotage dans la simulation

Environnement de pilotage

Les trois composantes logicielles

- Simulation numérique → boucle de calcul en temps
- Interface utilisateur → visualisation + interaction
- Système de communication → réalisation du couplage

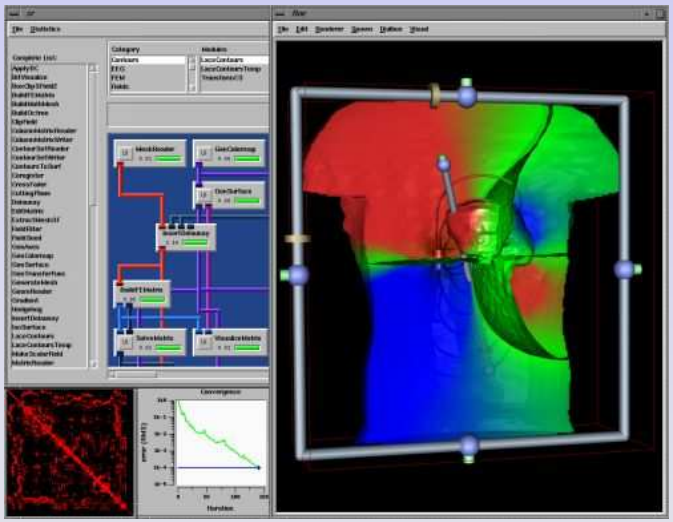


Monitoring/Steering **: observation/modification de l'état de la simulation à distance, en cours d'exécution**

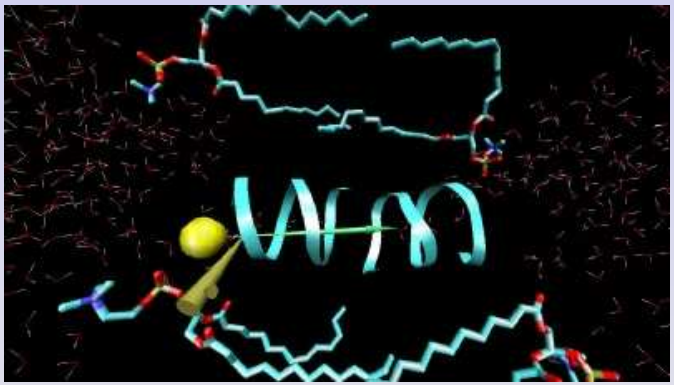
Plan

Les différentes approches

- Solution spécifique à une simulation (e.g. NAMD-VMD)
  → pilotage « sur mesure », non réutilisable
- Problem Solving Environment (PSE)
  → solutions dédiées à la construction d'une nouvelle application (SCIRun, COVISE)
- Solution générique : instrumentation
  → annotation d'un code source existant pour le rendre pilotable
  → amélioration de la qualité logicielle (découplage fort simulation/visualisation)



PSE SCIRun



NAMD-VMD

Modèle de pilotage

- Quel type de simulation peut-on piloter ?
  → séquentielle, multi-threads, parallèle
- Quelle représentation de la simulation dans l'environnement de pilotage ?
  → points d'interaction, boucles de calcul, graphes de tâches, objets distribués
- Quel paradigme pour le pilotage ?
  → communications bas-niveau, flux d'évènements, requêtes de pilotage

Environnement	**Simulation**	**Abstraction**	**Pilotage**
SCIRun (PSE)	séquentielle	modules	data-flow
COVISE (PSE)	séquentielle	modules	data-flow
VASE	séquentielle	graphe de tâches	data channels
CSE	séquentielle	points	read/write
Falcon	multi-threads	points	event stream
Progress/Magellan	multi-threads	points	event stream
PathFinder	SPMD	transactions	event stream
VIPER	SPMD	points	RPC
DAQV	SPMD	points	event stream
CUMULVS	SPMD	boucle de calcul	send/recv
MOSS	SPMD	objets distribués	RMI + event stream
DISCOVER	SPMD	objets distribués	RMI

Les différentes stratégies de coordination

Comment garantir la cohérence des opérations de pilotage en parallèle ?
→ ex. accès en lecture/écriture à des données distribuées de la simulation

- Cohérence spatiale : accès aux données autorisé uniquement en des endroits précis du code
- Cohérence temporelle : accès aux données au « même instant logique » pour tous les processus de la simulation

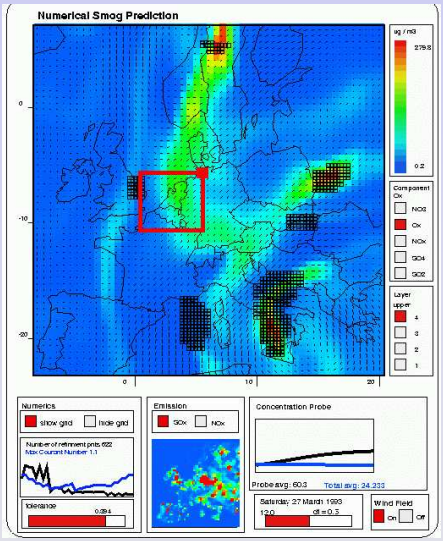
Quelques stratégies

- Event Reordering (Falcon, Progress/Magellan)
  → centralisation des évènements et tri *a posteriori*
- Synchronisation faible ou forte (CUMULVS, DAQV)
  → synchronisation plus ou moins forte à chaque itération

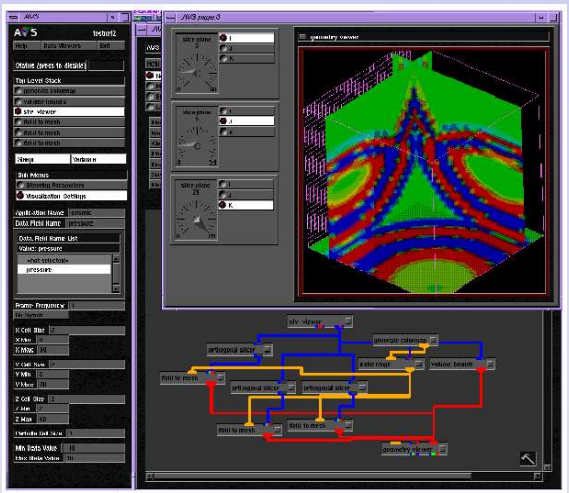
Stratégies induisant un surcoût important et systématique !

Quels outils pour piloter les simulations numériques ?

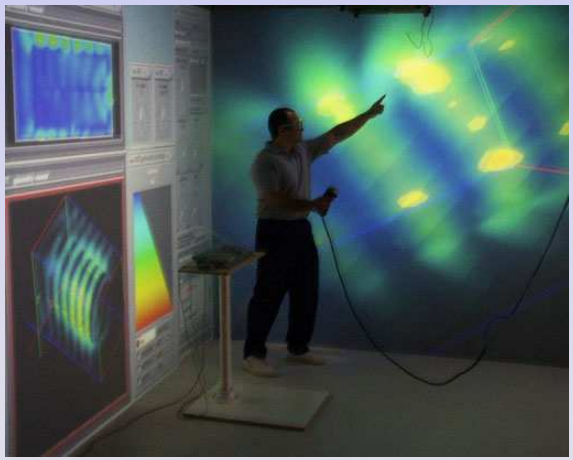
- Visualisation scientifique 2D/3D
  - → interface graphique « maison » (PGO de CSE)
  - → outils de visualisation professionnels (CUMULVS + AVS)
  - → immersion dans un CAVE (CAVEStudy, CUMULVS, CSE)
- Interactions de pilotage
  - → interactions simples (widgets 2D/3D)
  - → interactions par manipulation directe de l'image (Chatzikinos *et al.*)



PGO Editor de CSE (Mulder et al., 1998)



CUMULVS + AVS5



AVS/Express dans un CAVE (Kuo et al., 2000)

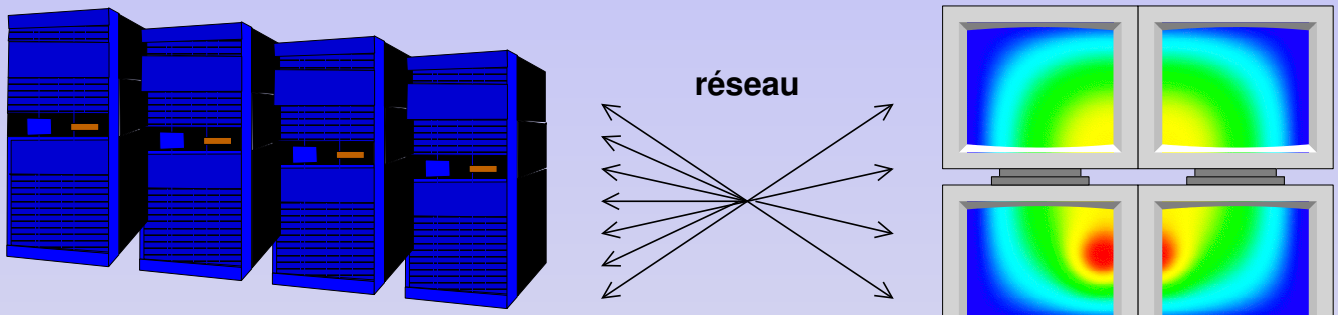
Plan

Positionnement : le pilotage $M \times N$

Piloter des simulations numériques parallèles en s'appuyant sur des outils de visualisation eux-mêmes parallèles

$\longrightarrow$ visualisation en ligne pour des simulations SPMD

$\longrightarrow$ gros volume de données, image à des résolutions élevées



Le pilotage $M \times N$ est vu comme un problème de couplage spécifique

$\longrightarrow$ couplage asymétrique, dynamique et non prévisible

Contributions

Les deux principaux problèmes que nous étudions :

- Le problème de la coordination efficace des opérations de pilotage
  → modèle de description des simulations parallèles : MHT
  → introduction système de datation
  → algorithme de planification des traitements pour garantir la cohérence

- Le problème de la redistribution de données complexes
  → modèle de description des données : objets complexes
  → étendre les travaux existants dans le cadre du pilotage
  → proposition de nouvelles approches pour la redistribution

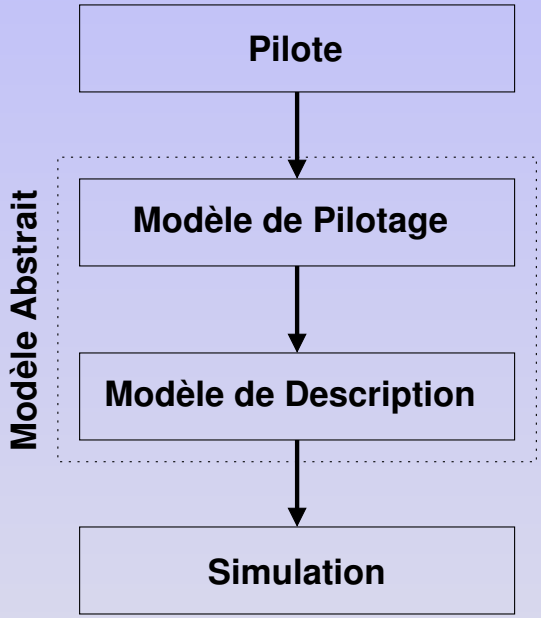
Conception et réalisation d'un environnement de pilotage EPSN
→ basé sur les technologies CORBA et VTK
→ développé en collaboration avec Michael Dussere (Ingénieur, INRIA Futurs)

Plan

Modèle pour un environnement de pilotage

Recherche d'une approche générique pour le pilotage de simulations SPMD

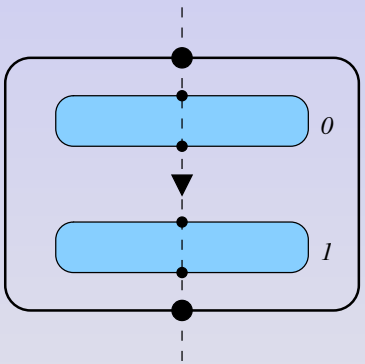
- Modèle de description
  → basé sur l'instrumentation du code source
  → structure du programme : modèle hiérarchique en tâches (MHT)
  → données distribuées : objets complexes
- Modèle de pilotage
  → pilotage par des requêtes
  → association des interactions possibles aux tâches du MHT



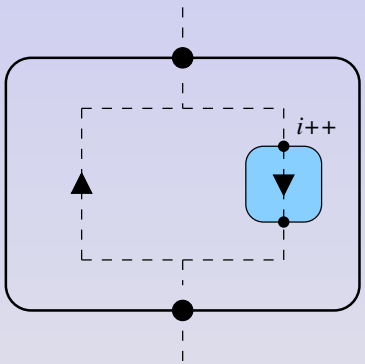
Le modèle doit indiquer à l'environnement de pilotage où, quand et comment interagir de manière cohérente avec le code de simulation.

Description des programmes structurés séquentiels ou SPMD

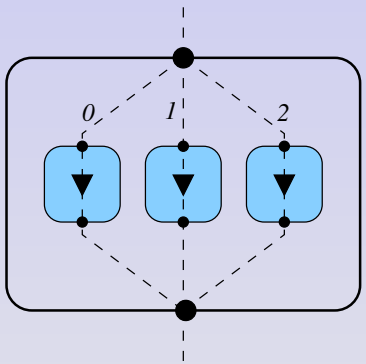
- Modèle simple basé sur un arbre de tâches
  → tâche contenant des sous-tâches
- Inspiré des travaux de Polychronopoulos *et al.* sur les HTGs
- Quatre types de tâches (simple, boucle, conditionnelle, point)
  → capturer le flot d'exécution
- Description à grain moyen du code de simulation
  → annoter dans le source uniquement les tâches pertinentes



(a) tâche composée



(b) tâche en boucle

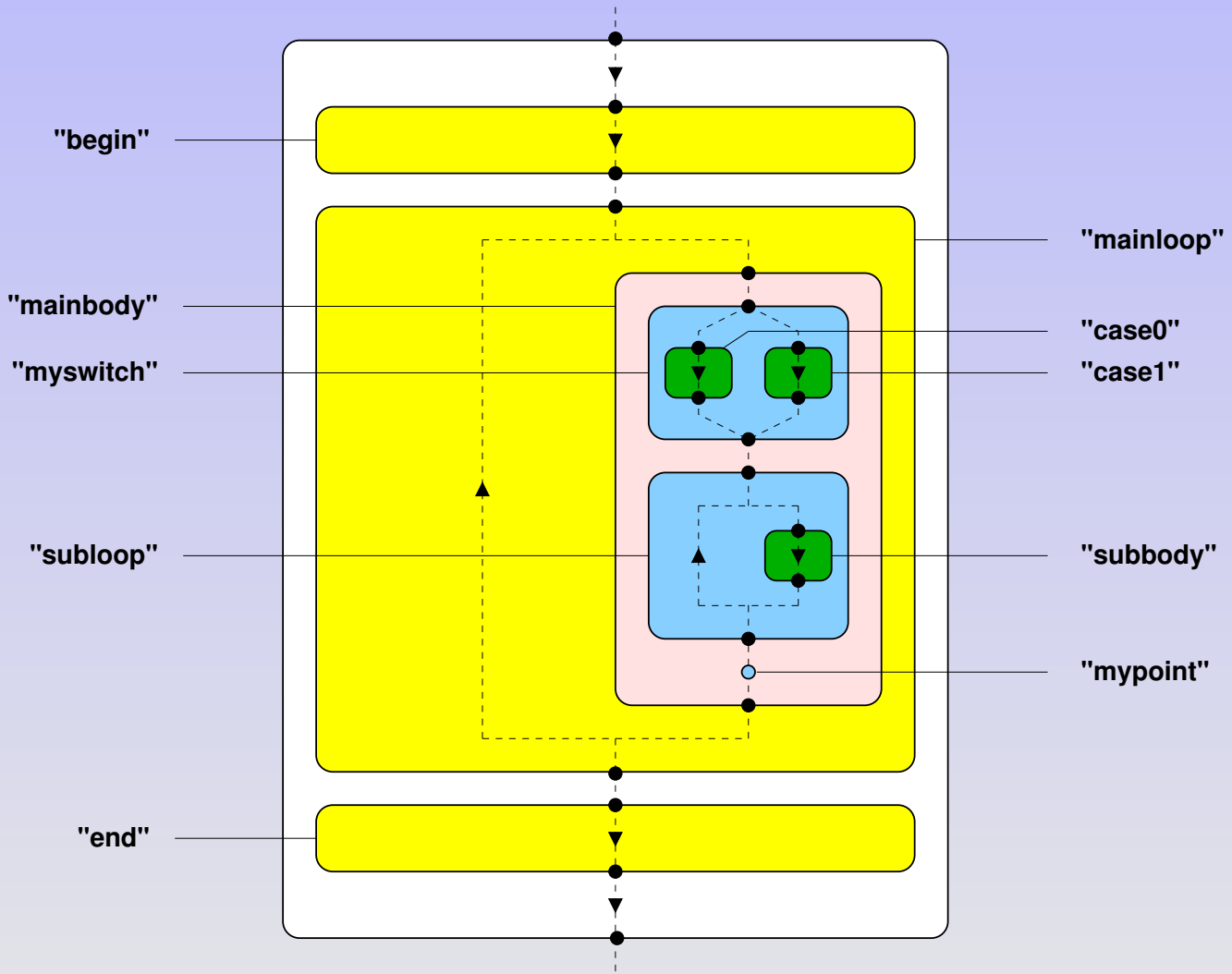


(c) tâche conditionnelle

(d) tâche en point

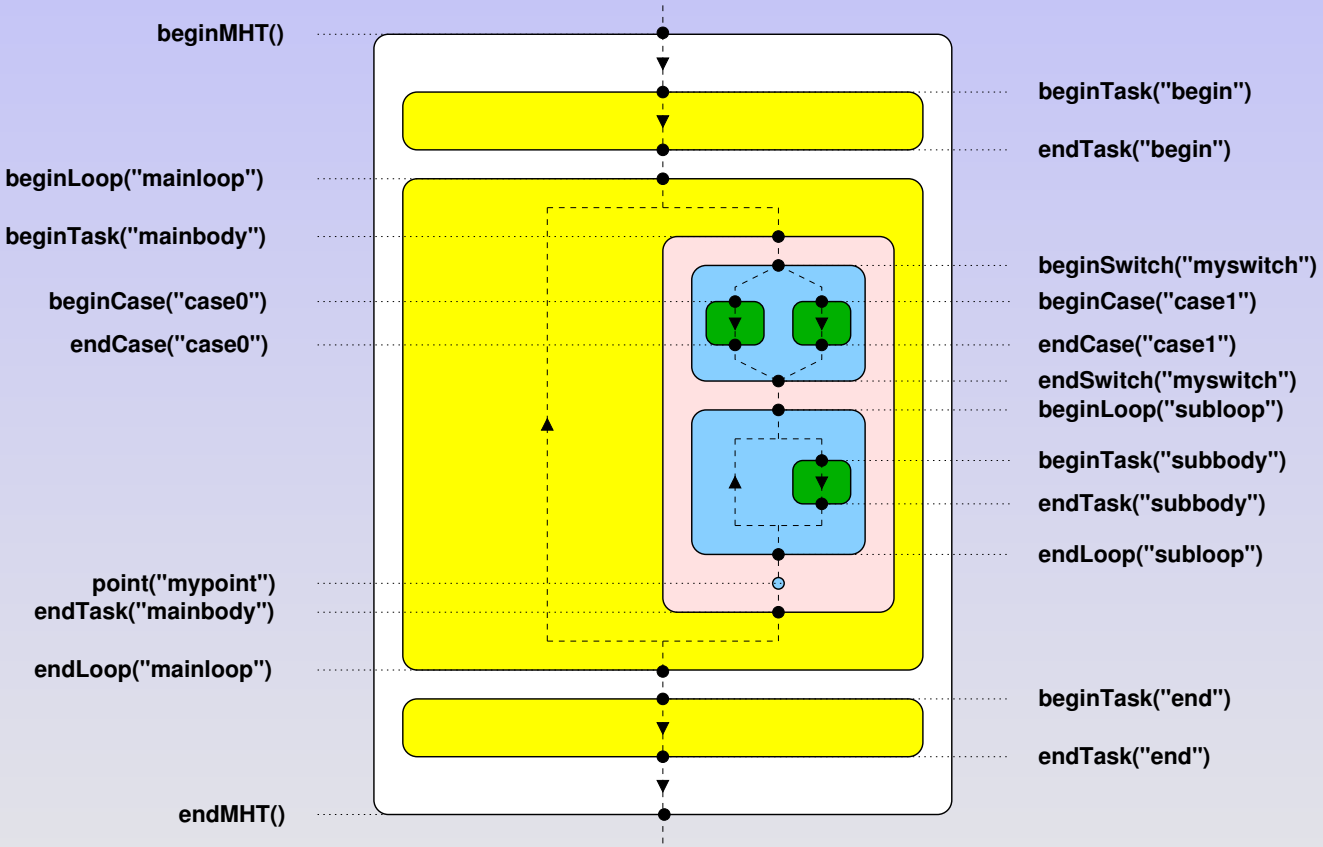
Exemple de MHT

Exemple avec deux boucles imbriquées, une conditionnelle et un point d'interaction



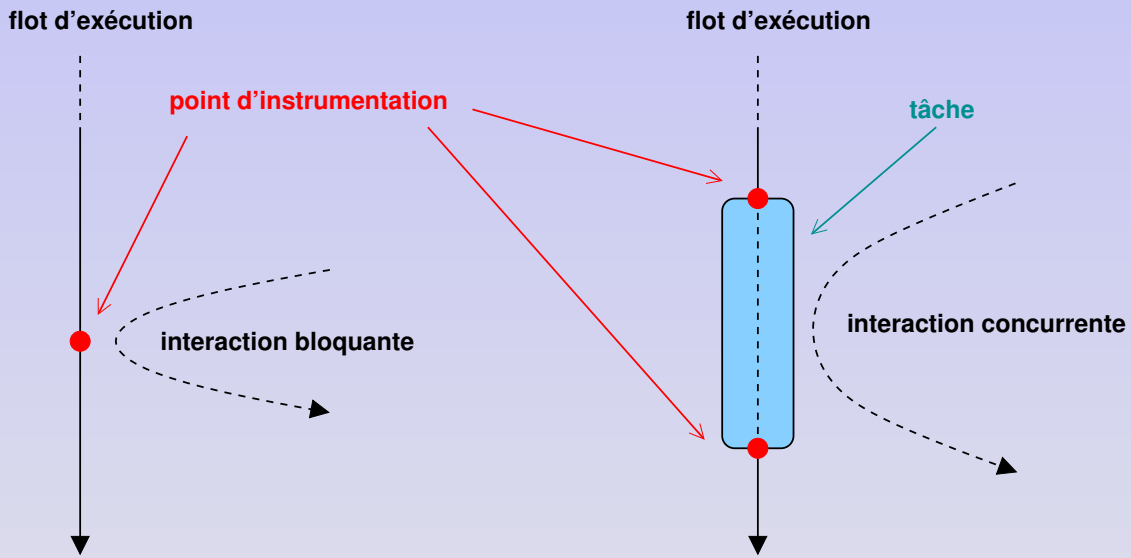
Construction du MHT par l'utilisateur

- Repérer dans le code le début et la fin de chaque tâche
  → points d'instrumentation (ex. *beginTask/endTask*)
- Description XML du MHT
  → connaissance statique et vision hiérarchique



Mécanismes d'interaction

- Interaction bloquante sur un point du MHT
  → surcoût maximal dans la simulation
- Interaction concurrente à l'exécution d'une tâche du MHT
  → fin de tâche bloquante si interaction non terminée
  → recouvrement possible des interactions (diminution du surcoût)



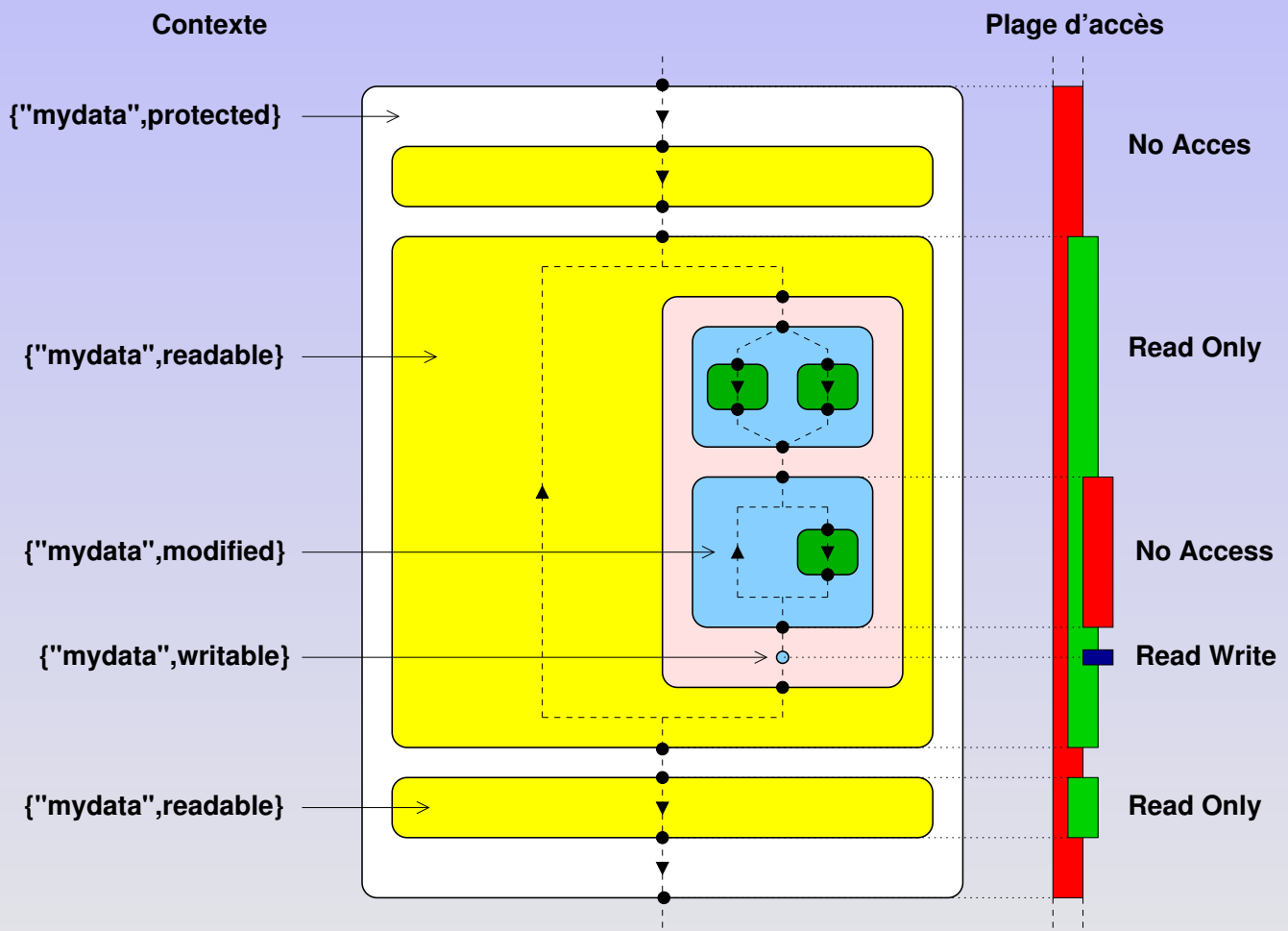
(a) point d'interaction

(b) plage d'interaction

Exploitation de plage d'accès aux données pour recouvrir les transferts !

Plage d'accès aux données

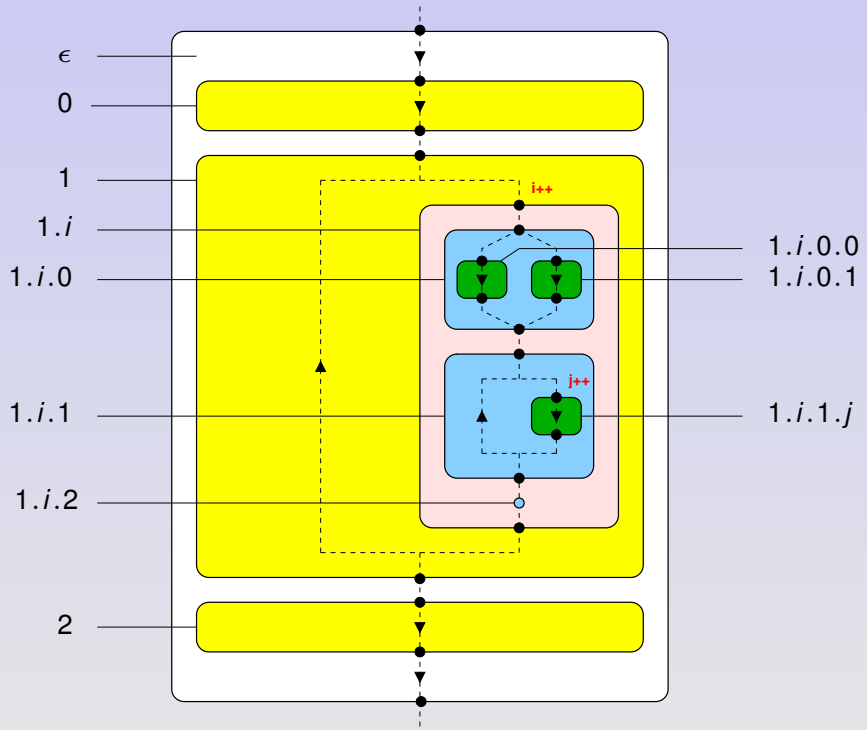
- Contexte d'accès associé aux tâches : *readable, writable, modified, protected*
- Configuration des plages d'accès via un fichier XML décrivant le MHT



Système de datation associé au MHT

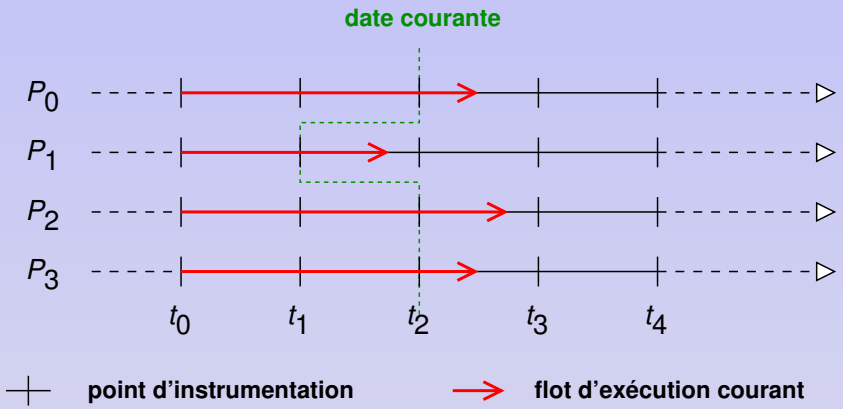
Suivre précisément l'évolution des calculs sur le MHT

- Dates de tâche
  → concaténation d'indices relativement à la hiérarchie
  → indice : compteur d'itérations pour les boucles, branche de la conditionnelle, ...
- Dates de point
  → associée aux points d'instrumentation du MHT
  → date de tâche + position en début ou en fin de tâche
  → relation d'ordre stricte et totale pour comparer les dates



Modèle d'exécution pour des simulations SPMD

- Chaque processus parcourt le même MHT
  → exécution *a priori* non synchronisée de la simulation
- Date courante d'un processus
  → dernière date de point rencontrée par le flot d'exécution



Hypothèse d'un parcours SPMD strict du MHT

Tous les processus P_i rencontrent la même séquence de points d'instrumentation aux dates t_0, t_1, *etc.*

→ même nombre d'itérations dans les boucles
→ même choix de branche pour les conditonnelles

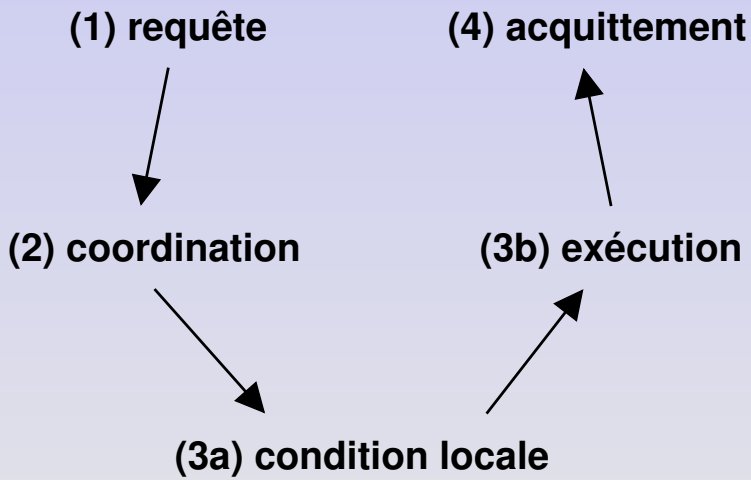
Système de pilotage par les requêtes

Modèle de pilotage par les requêtes

- Les requêtes simples
  → get, put, action, play/step/stop
- Requêtes permanentes
  → envoi périodique (getp), actions répétées
- Requêtes complexes
  → composition de requêtes simples (multi-get, actions paramétrées, ...)

Le cycle de vie des requêtes

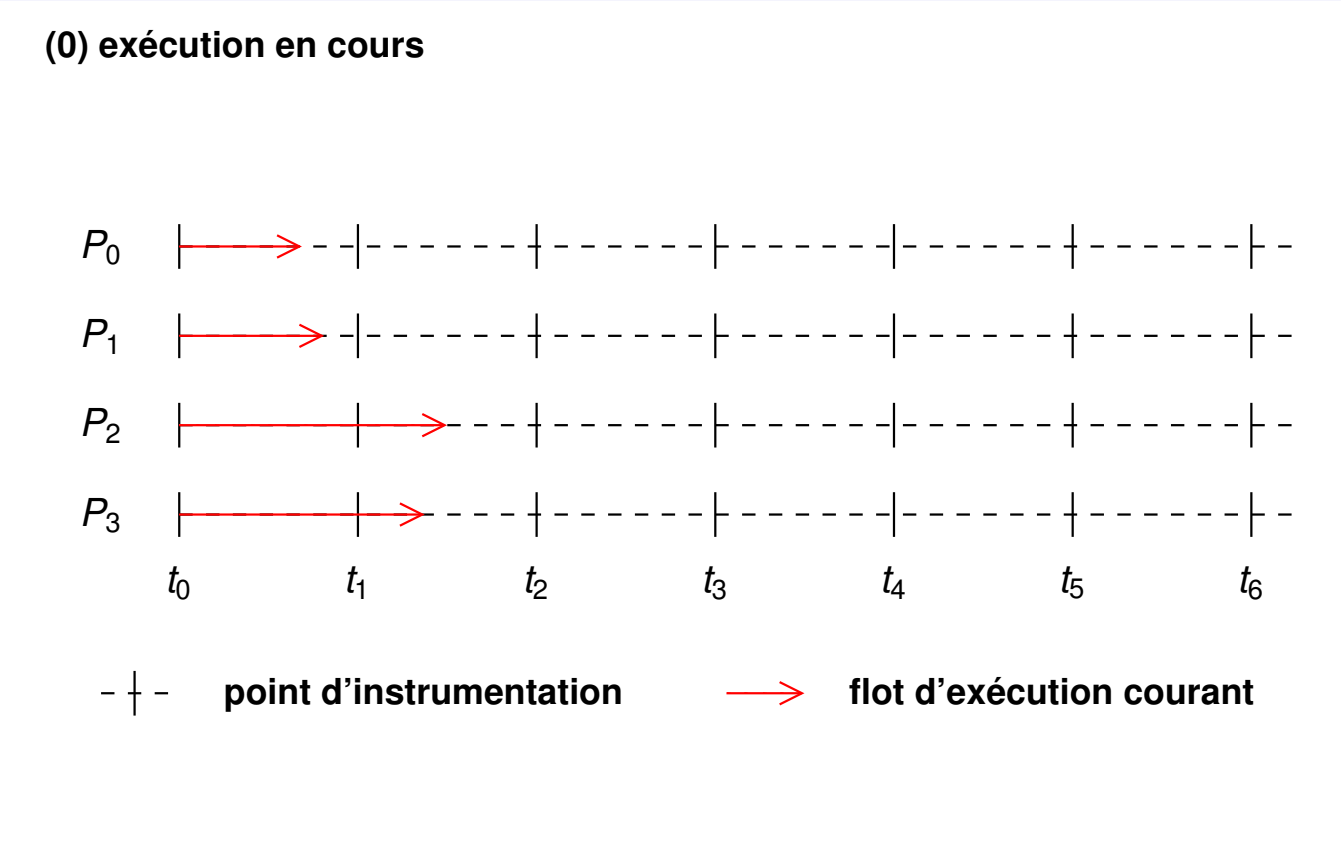
1. Envoi de la requête cliente
2. Coordination du traitement de la requête
3. Traitement de la requête
   → évaluation d'une « condition locale »
   → exécution de la requête
4. Acquittement marquant la fin du traitement



Coordination : algorithme de planification

Planifier une date de traitement afin de garantir la cohérence temporelle

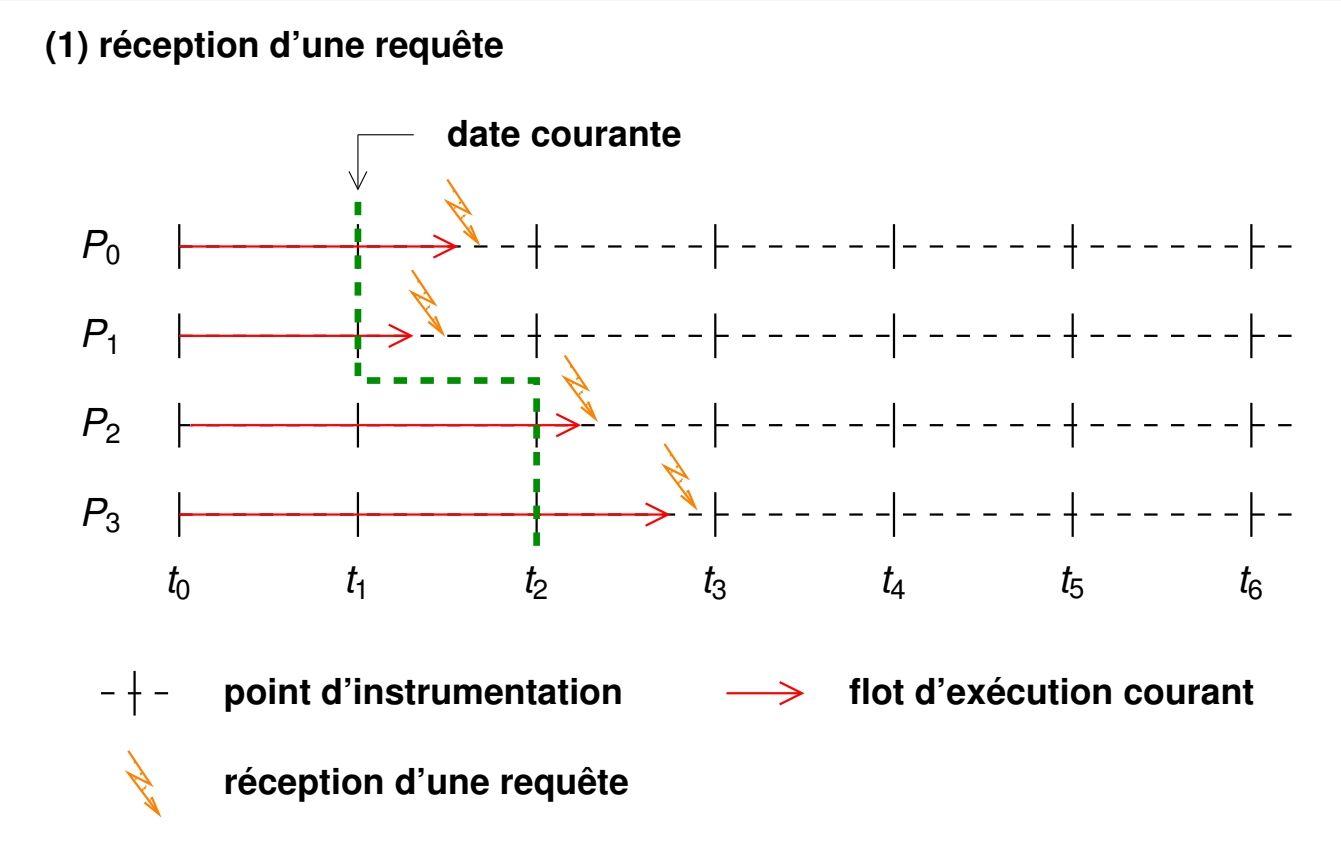
- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse



Coordination : algorithme de planification

Planifier une date de traitement afin de garantir la cohérence temporelle

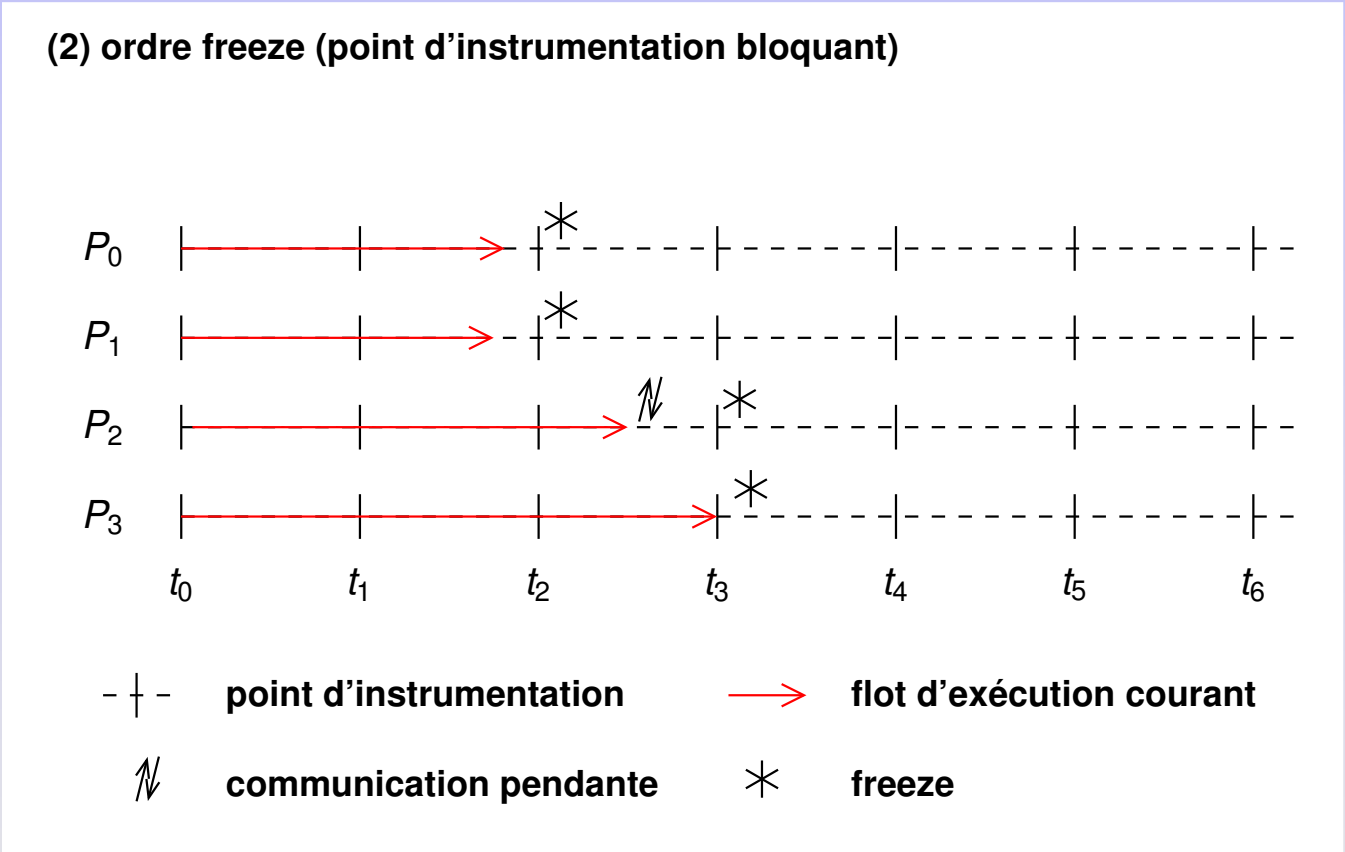
- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse



Coordination : algorithme de planification

Planifier une date de traitement afin de garantir la cohérence temporelle

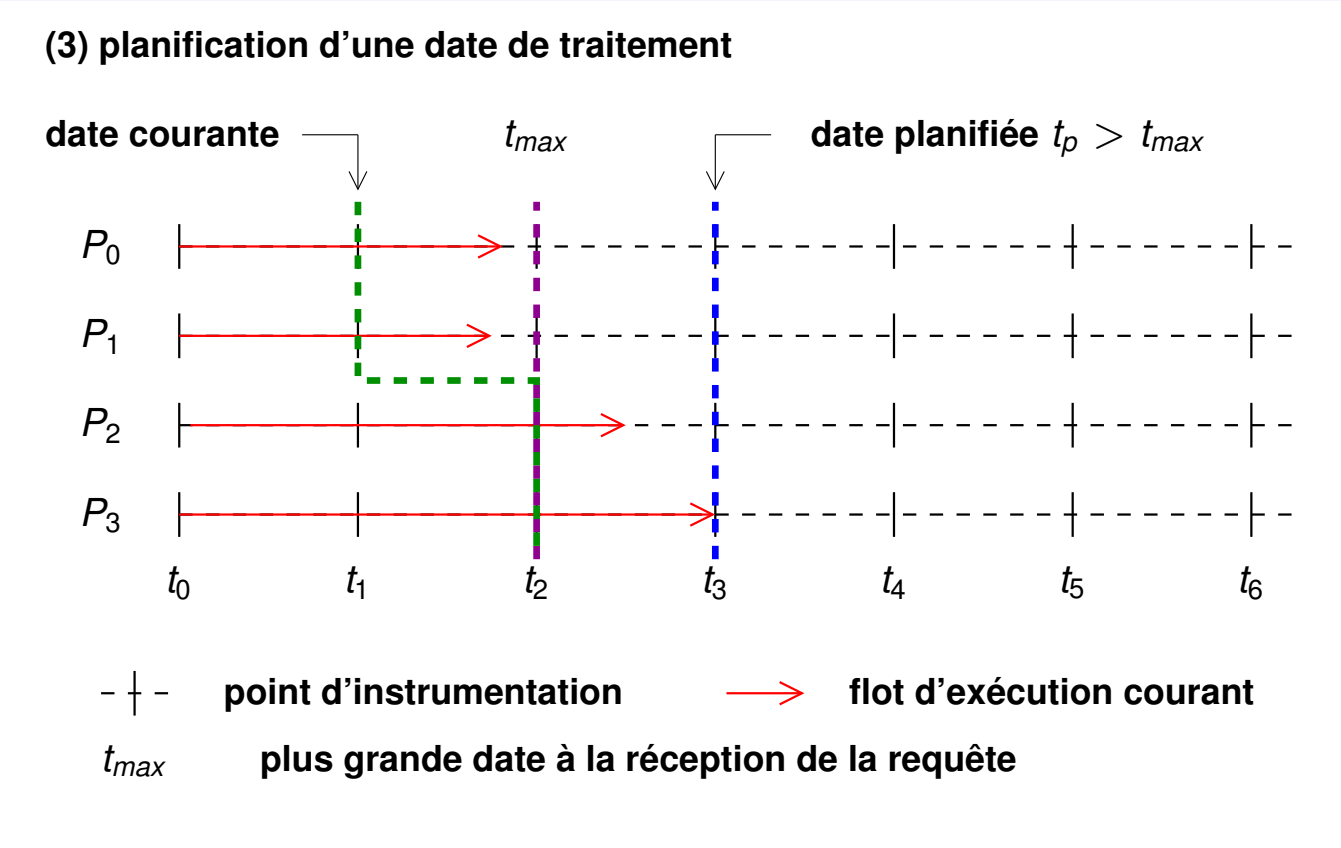
- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse



Coordination : algorithme de planification

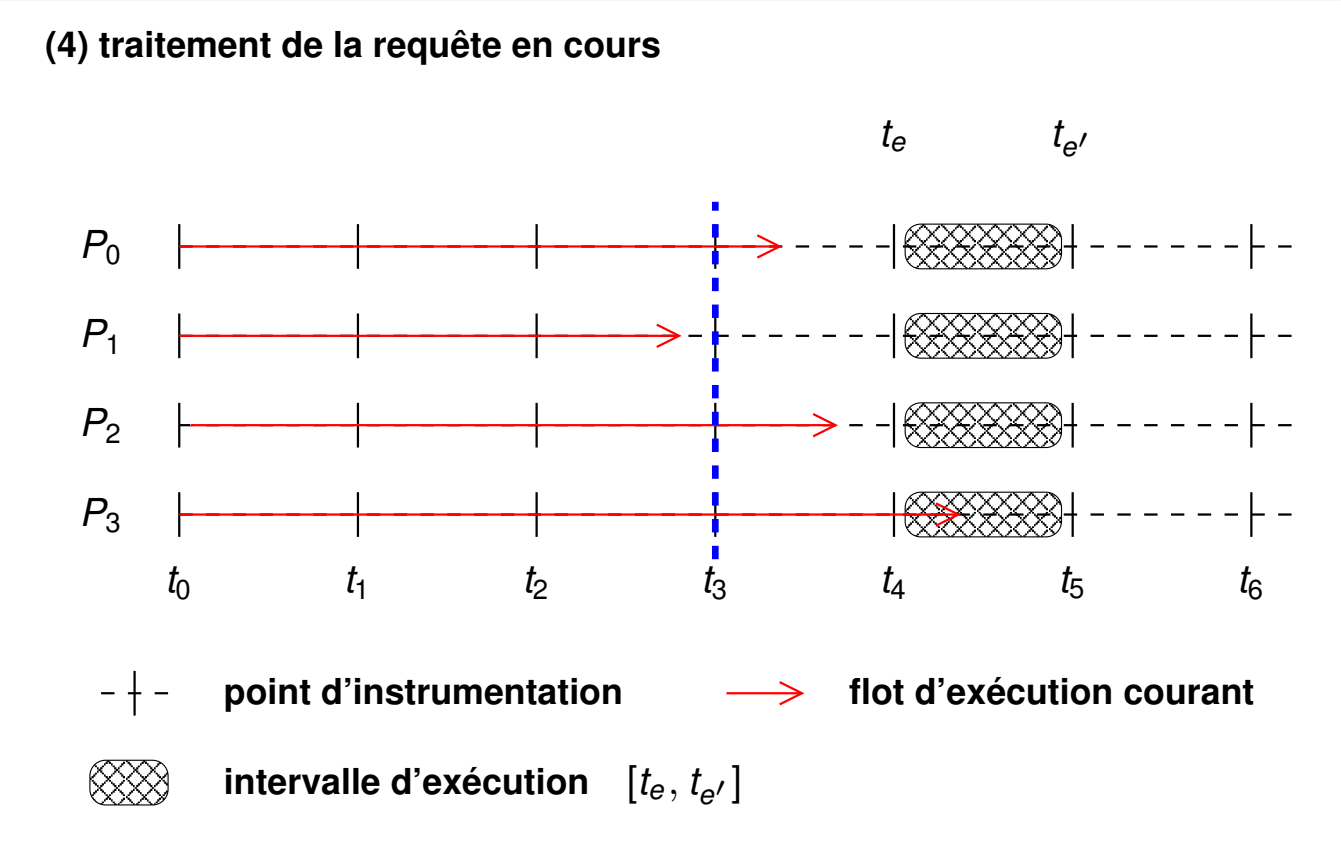
Planifier une date de traitement afin de garantir la cohérence temporelle

- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse



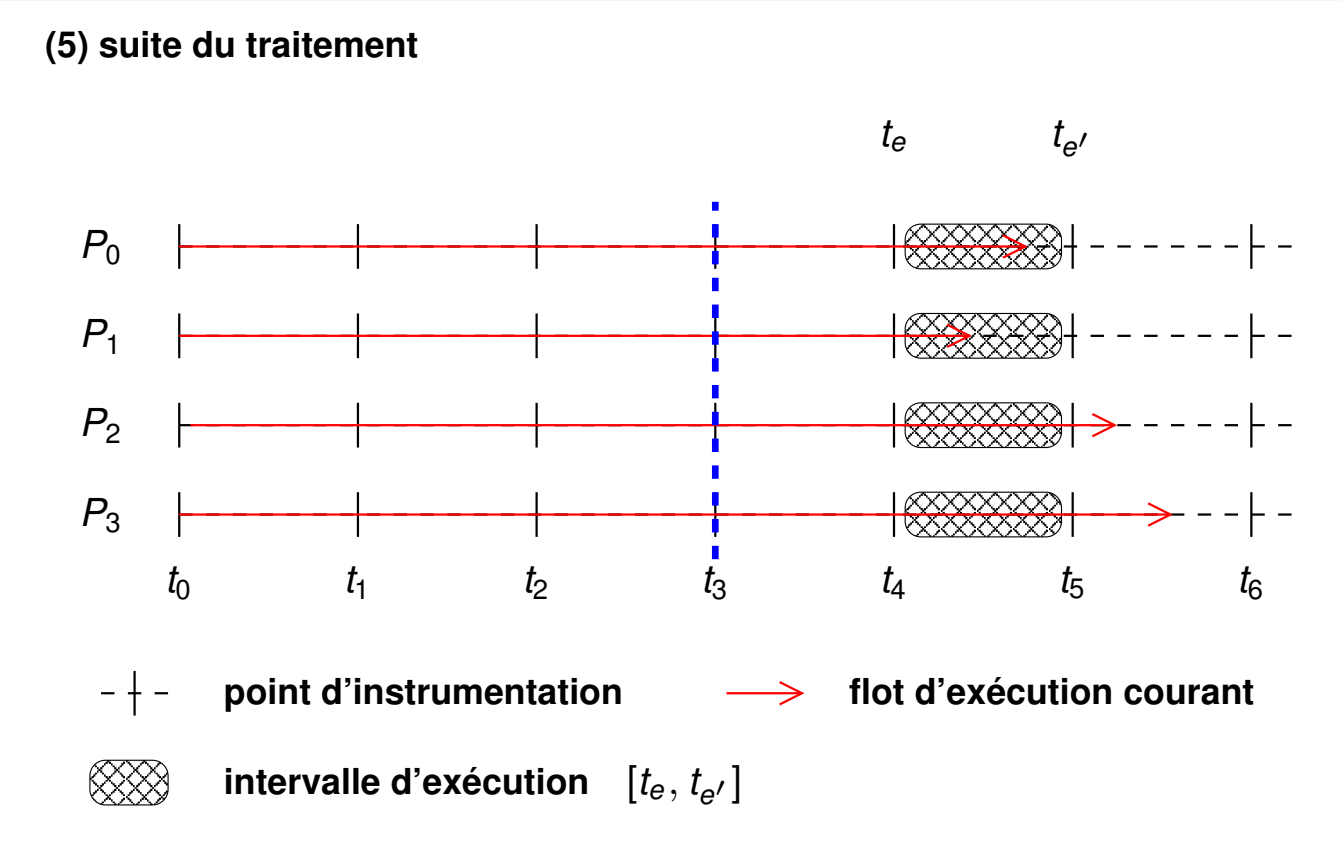
Planifier une date de traitement afin de garantir la cohérence temporelle

- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse



Planifier une date de traitement afin de garantir la cohérence temporelle

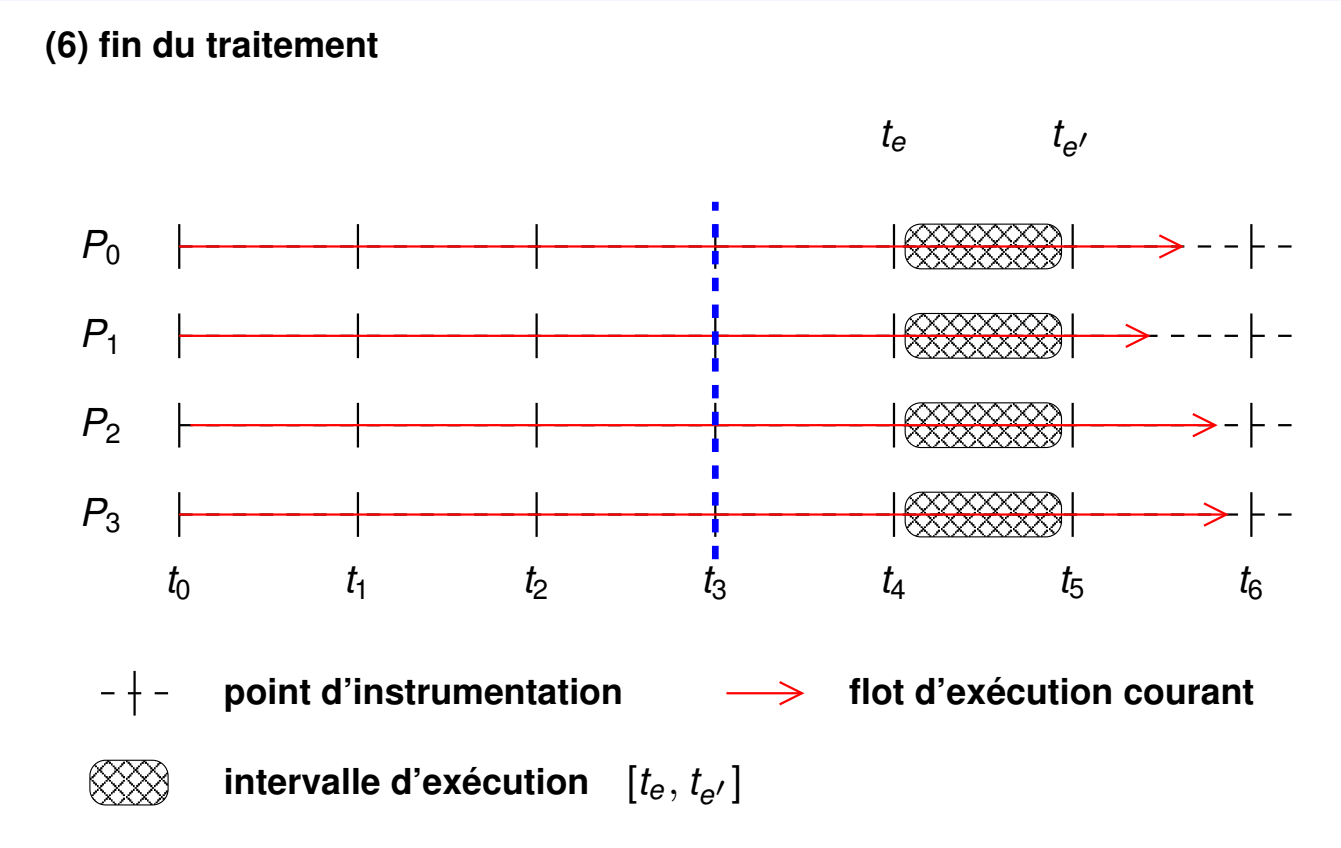
- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse



Coordination : algorithme de planification

Planifier une date de traitement afin de garantir la cohérence temporelle

- Difficultés : simulation parallèle non synchronisée, requêtes non prévisibles
- Objectif : établir une date commune en évitant toute synchronisation coûteuse

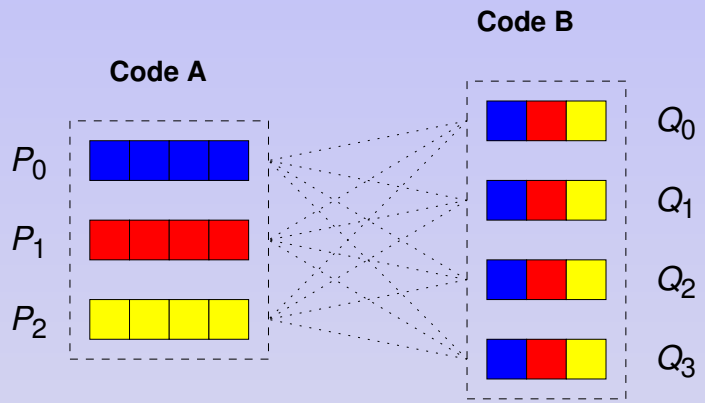


Plan

Le problème de la redistribution des données

Travaux dans le cadre plus général du couplage de codes

- Deux ensembles de processeurs disjoints P et Q (de taille M et N)
- Un même ensemble de données partagé entre deux codes A et B
- Nombre de processus M et N différents, distributions différentes

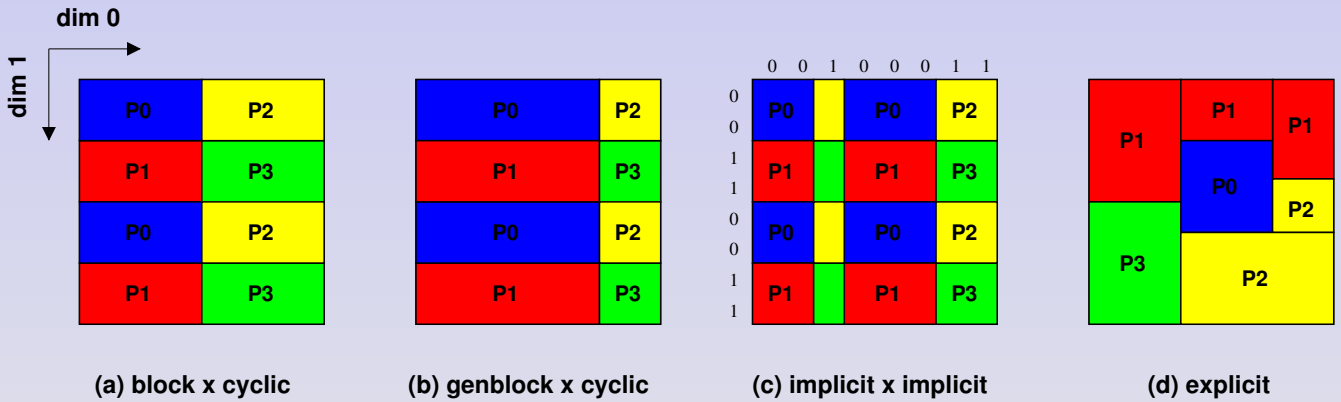


Les étapes de la redistribution

1. Description des données (distribution, stockage)
2. Génération des messages
3. Ordonnancement (étapes de communication)
4. Communication (flux parallèles)

Travaux existants

- Tableaux multi-dimensionnels denses
  → distribution régulière bloc-cyclique : HPF, ScaLAPACK ($M = N$)
  → distribution explicite : PAWS, CCA M×N
- Principe de linéarisation
  → approche très générique, indépendante structure de données
  → Meta-Chaos/InterComm, MPI I/O M×N
- Solutions dédiées plus spécifiques (couplage multi-physiques)
  → MCT (grilles), MpCCI (interpolation grilles/maillages)

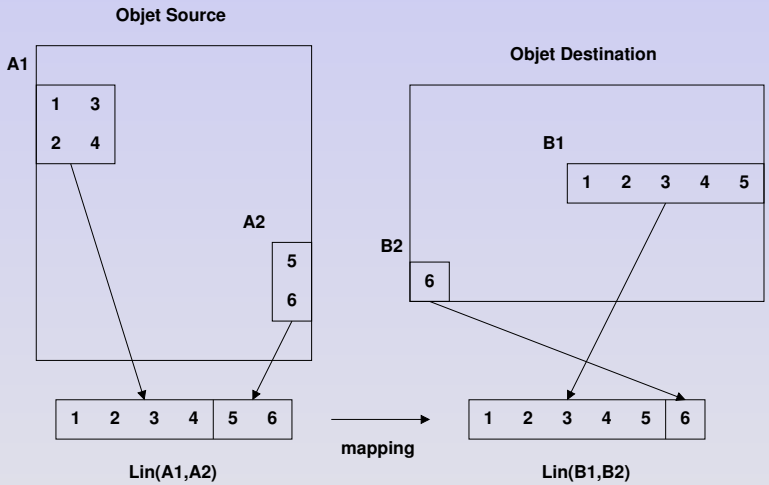


(a) block x cyclic (b) genblock x cyclic (c) implicit x implicit (d) explicit

CCA M×N

Travaux existants

- Tableaux multi-dimensionnels denses
  → distribution régulière bloc-cyclique : HPF, ScaLAPACK ($M = N$)
  → distribution explicite : PAWS, CCA M×N
- Principe de linéarisation
  → approche très générique, indépendante structure de données
  → Meta-Chaos/InterComm, MPI I/O M×N
- Solutions dédiées plus spécifiques (couplage multi-physiques)
  → MCT (grilles), MpCCI (interpolation grilles/maillages)

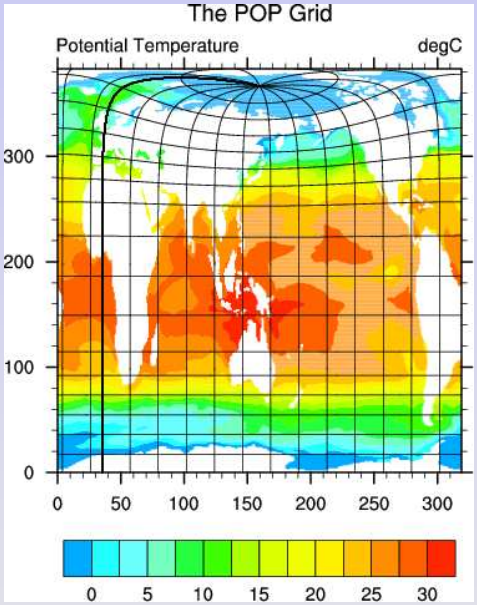


Principe de linéarisation.

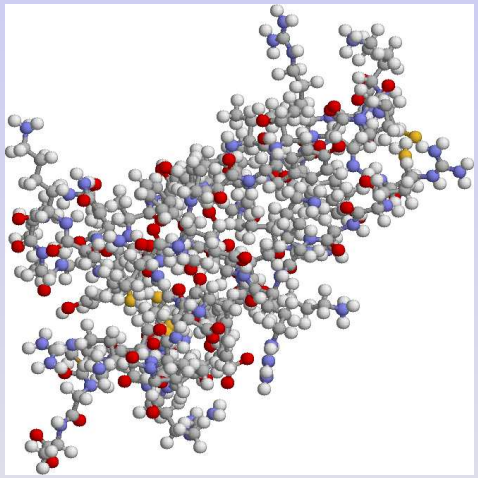
Motivations

Besoin d'étendre les travaux sur la redistribution pour...

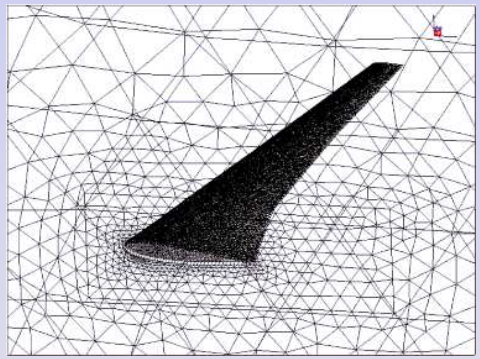
- Supporter des distributions de données plus irrégulières
  → distributions par bloc quelconques (denses ou creuses)
- Redistribuer des objets plus complexes
  → grilles, ensembles de particules, molécules, maillages non structurés...
- Etre indépendant de la couche de communication
  → réutilisabilité, redistribution symbolique



Décomposition spatiale dans POP



Molécule dans NAMD

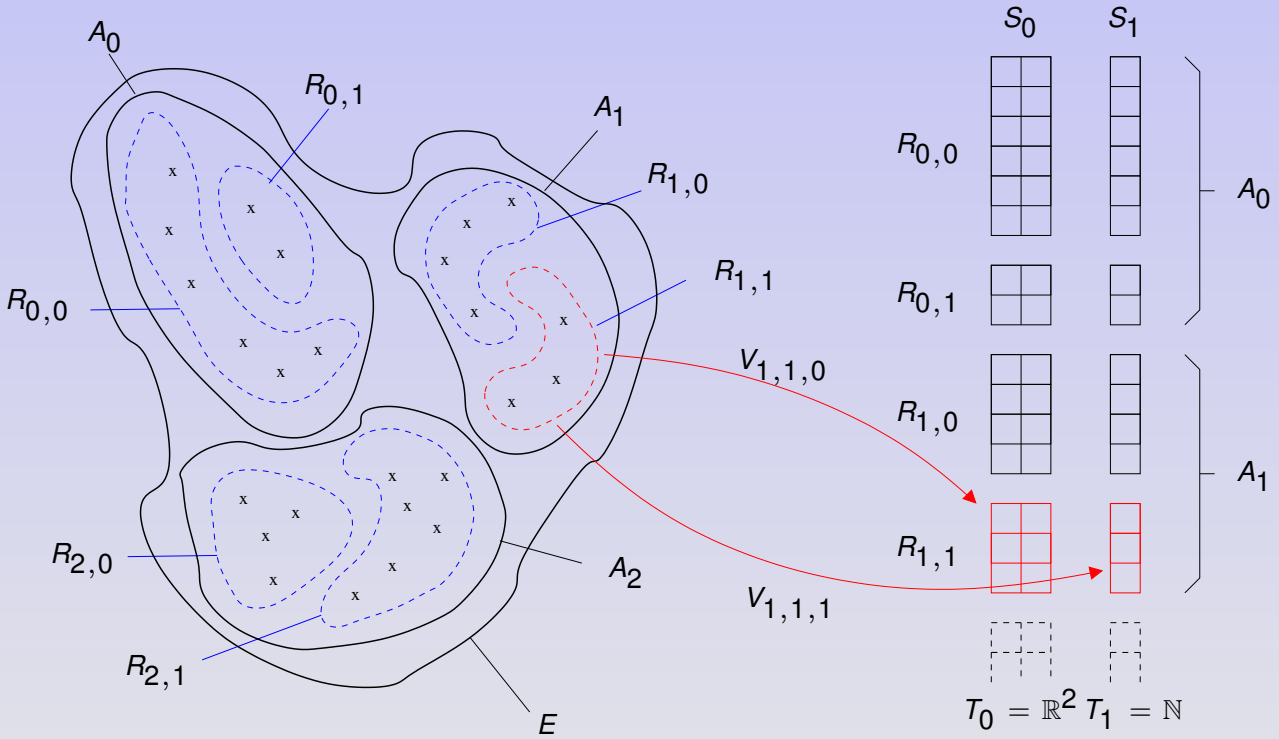


Maillage d'une aile d'avion.

Objets complexes

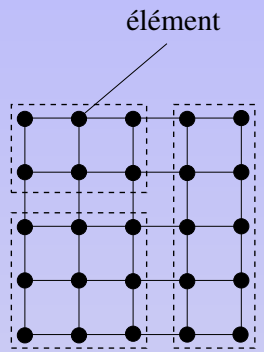
Décrire les objets physiques manipulés par les codes de simulation

- Ensemble d'éléments totalement ordonnés distribués sur M processeurs
- Éléments regroupés en régions logiques (sous-ensemble d'éléments, ordre local)
- Séries de données associées aux éléments (stockage en *stride*)

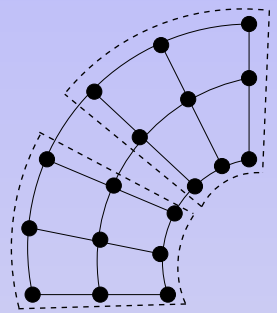


Les différentes classes d'objets complexes

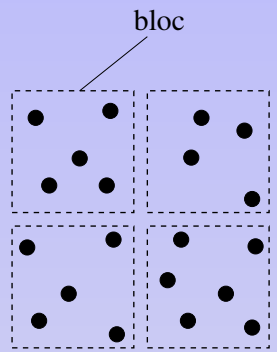
grilles structurées, boîtes d'atomes, ensembles de particules, maillages non structurés



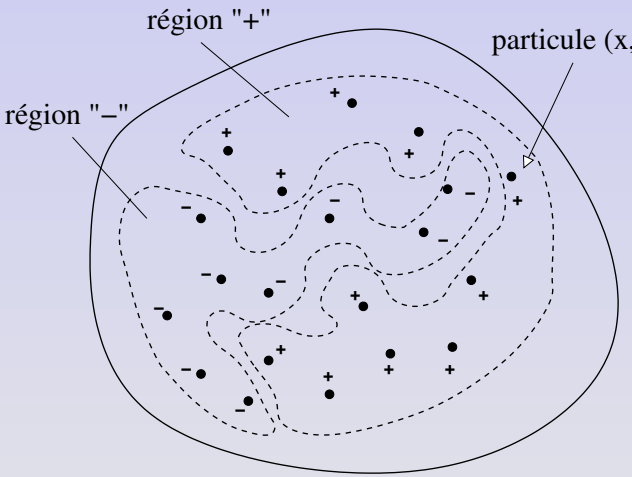
(a) grille régulière



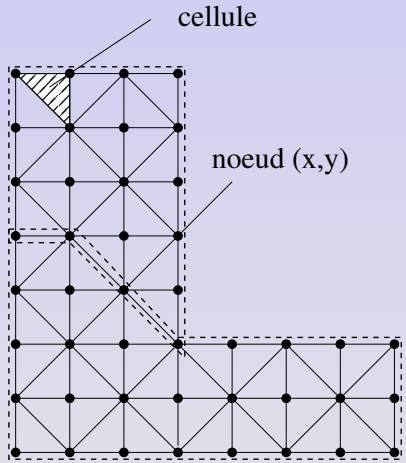
(b) grille irrégulière



(c) boîtes d'atomes



(d) ensemble de particules



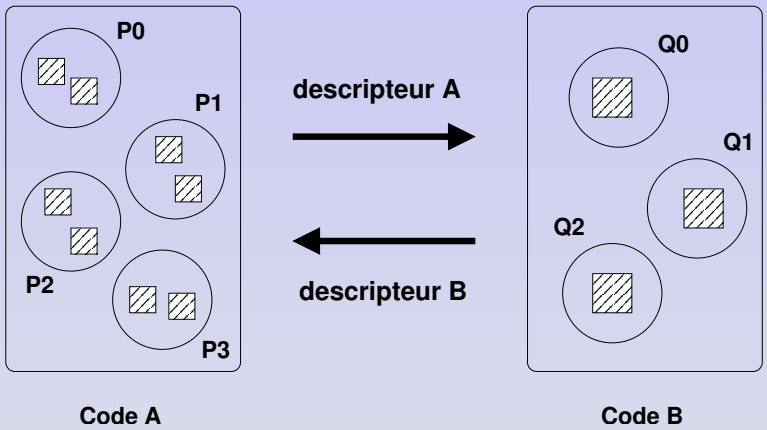
(e) maillage non structuré

Algorithme de redistribution parallèle

Initialisation

- Chaque processeur connaît la description de l'objet local.
- Il a besoin de connaître l'ensemble de la distribution distante.
- Il n'a pas besoin de connaître l'ensemble de la distribution locale !

Contexte du couplage de codes



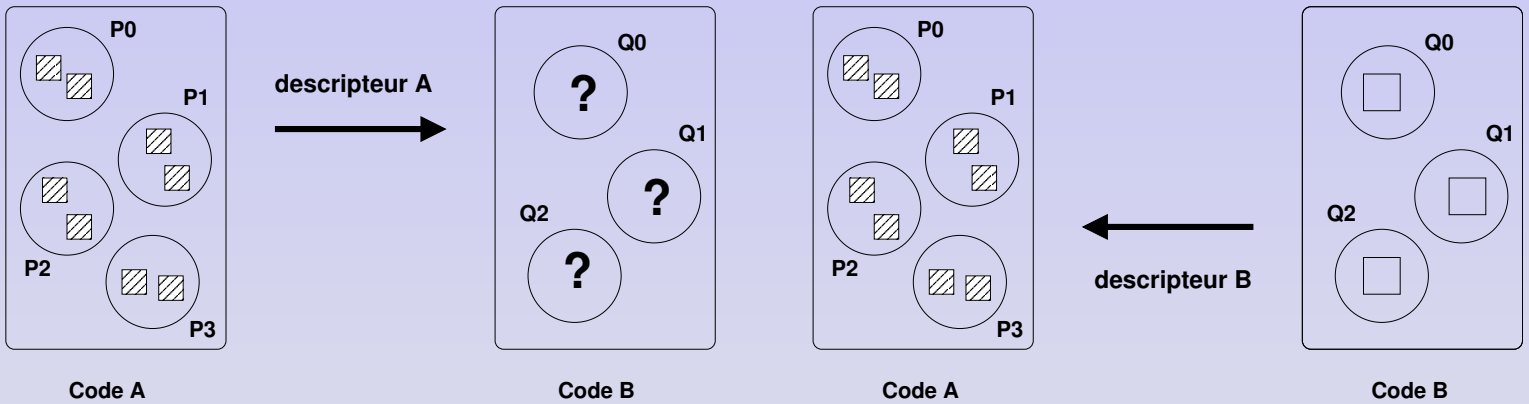
Approche spatiale (algorithme symétrique)

Algorithme de redistribution parallèle

Initialisation

- Chaque processeur connaît la description de l'objet local.
- Il a besoin de connaître l'ensemble de la distribution distante.
- Il n'a pas besoin de connaître l'ensemble de la distribution locale !

Contexte du pilotage

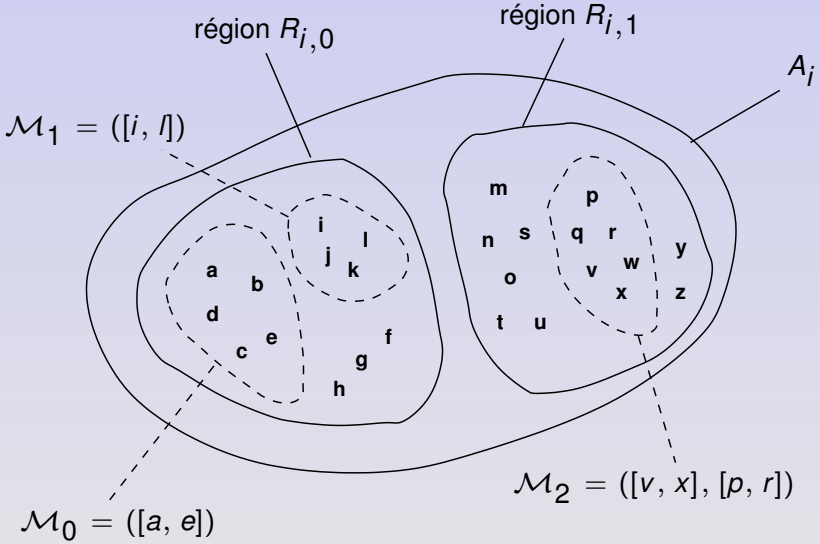


Choix d'une distribution pour le code B (visualisation) :
1) on se ramène au contexte symétrique du couplage
2) approche placement (algorithme asymétrique)

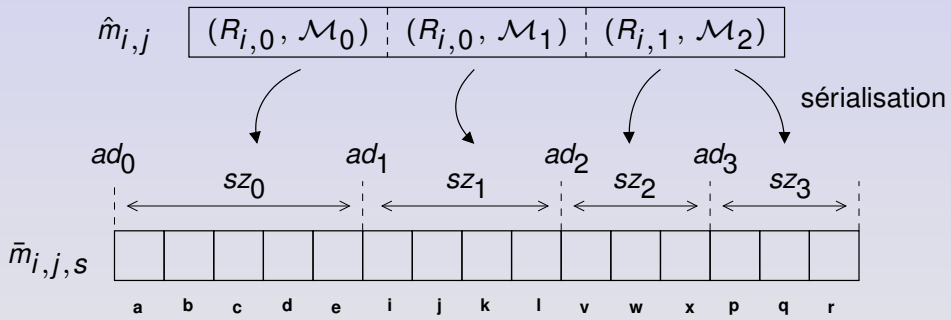
Génération message

Pour chaque processeur P_i, faire en parallèle :
Pour chaque processeur distant Q_j, faire :

- Calculer le message symbolique $\hat{m}_{i,j}$ de P_i vers Q_j (extraction)
- Pour chaque série de données S_s, faire :
  Générer le message physique $\bar{m}_{i,j,s}$ à partir de $\hat{m}_{i,j}$ (sérialisation)

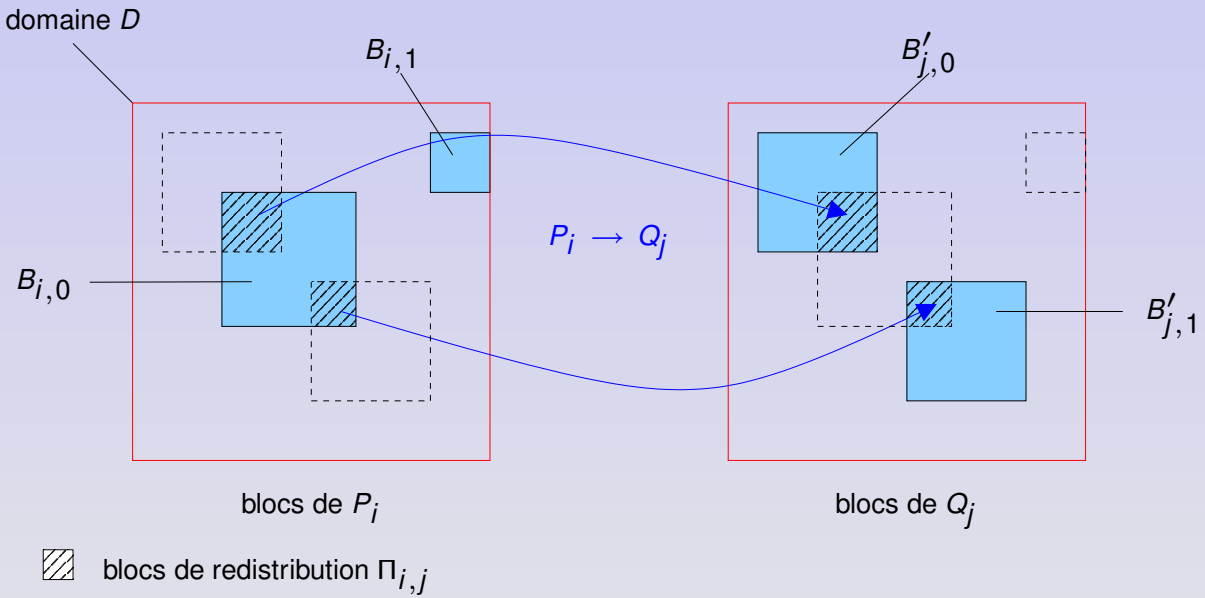


$$\hat{m}_{i,j} = ([a, e], [i, l], [v, x], [p, r])$$



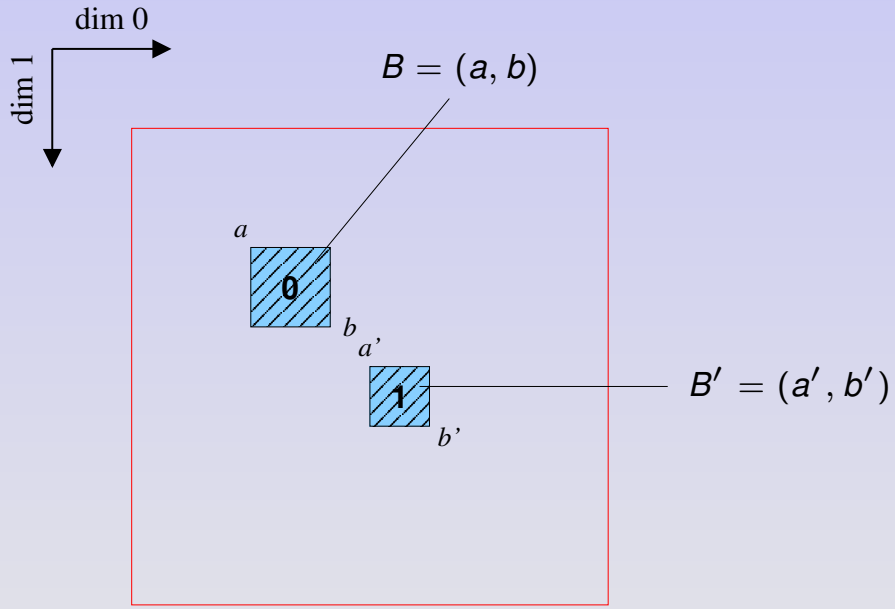
Application aux objets distribués par blocs : grilles structurées, boîtes d'atomes

1. Calcul de l'ensemble des blocs de redistribution ($\Pi_{i,j}$) par intersection géométrique des blocs de P_i et de Q_j
2. Tri des blocs de redistribution dans un ordre canonique
3. Génération du message $\hat{m}_{i,j}$ par extraction des éléments associés aux blocs de redistribution



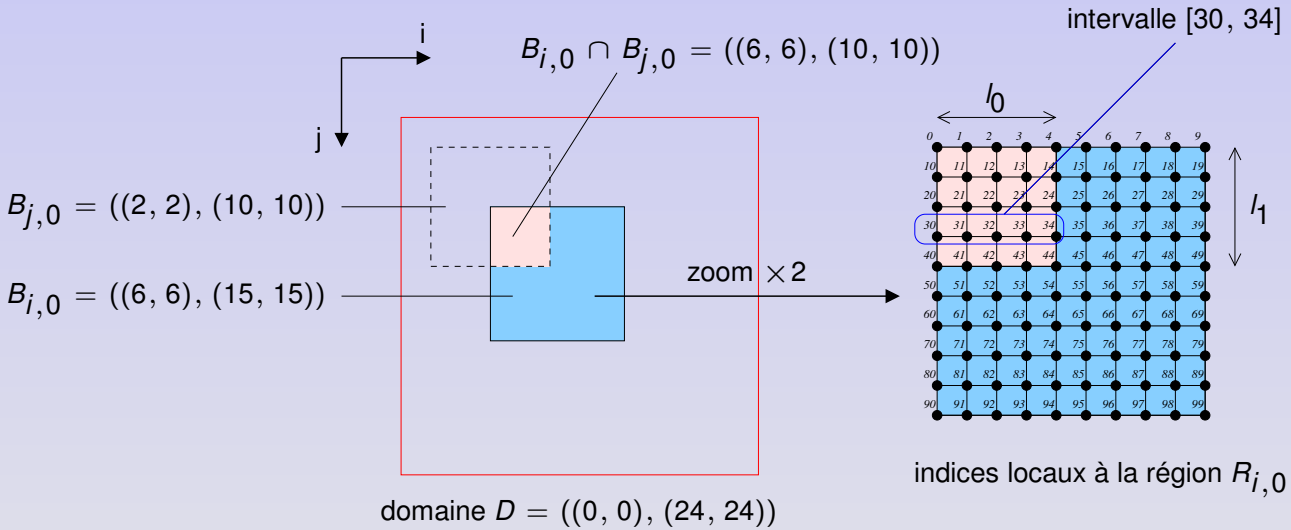
Application aux objets distribués par blocs : grilles structurées, boîtes d'atomes

1. Calcul de l'ensemble des blocs de redistribution ($\Pi_{i,j}$) par intersection géométrique des blocs de P_i et de Q_j
2. Tri des blocs de redistribution dans un ordre canonique
3. Génération du message $\hat{m}_{i,j}$ par extraction des éléments associés aux blocs de redistribution



Application aux objets distribués par blocs : grilles structurées, boîtes d'atomes

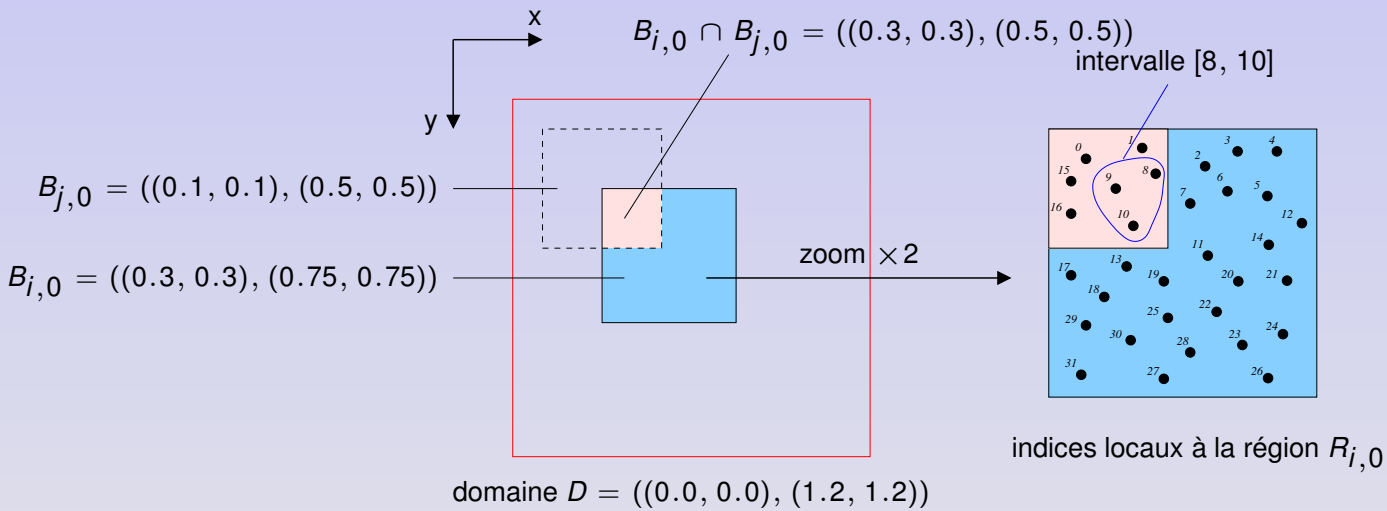
1. Calcul de l'ensemble des blocs de redistribution ($\Pi_{i,j}$) par intersection géométrique des blocs de P_i et de Q_j
2. Tri des blocs de redistribution dans un ordre canonique
3. Génération du message $\hat{m}_{i,j}$ par extraction des éléments associés aux blocs de redistribution



extraction dans le cas des grilles structurées

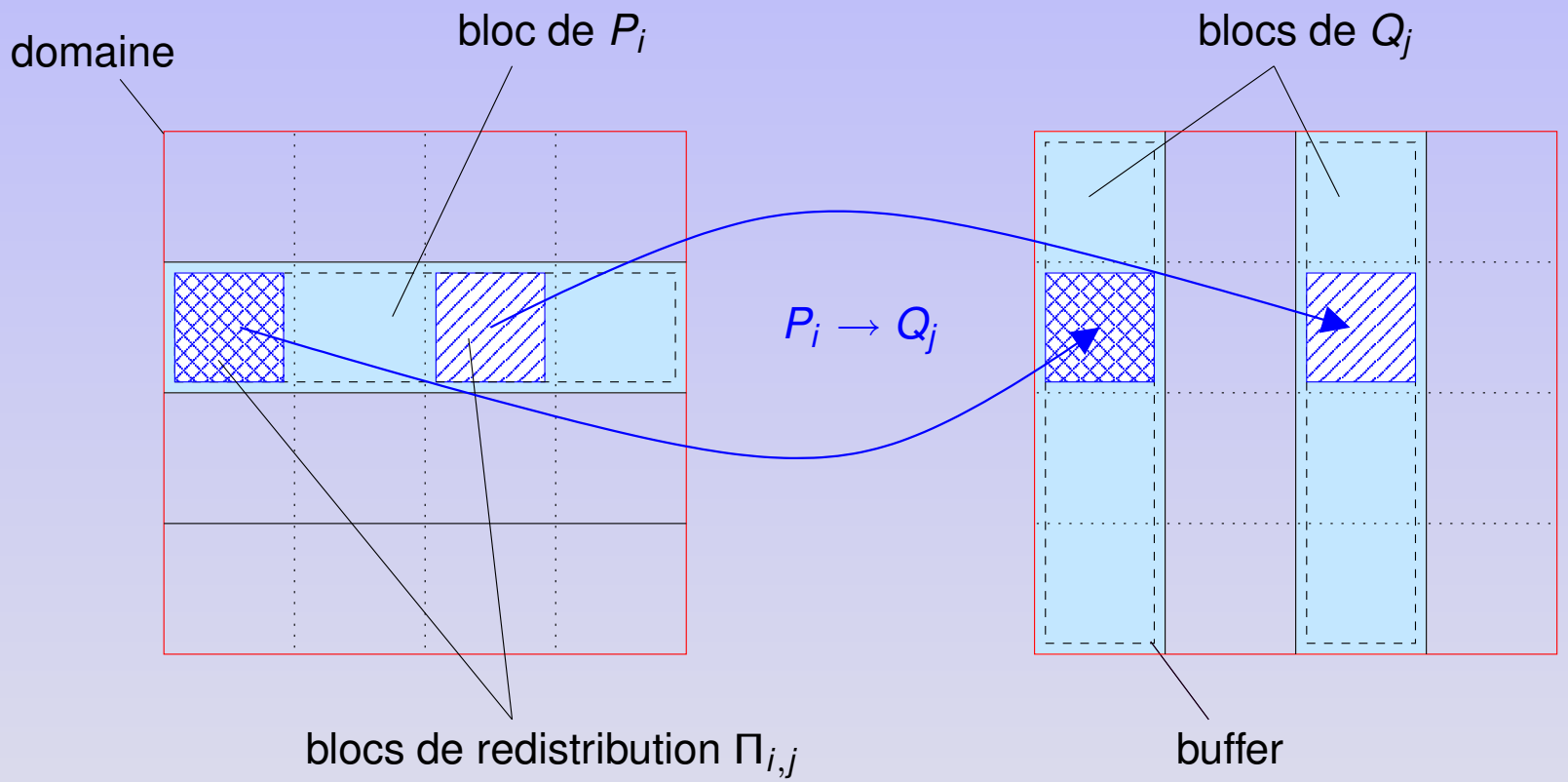
Application aux objets distribués par blocs : grilles structurées, boîtes d'atomes

1. Calcul de l'ensemble des blocs de redistribution ($\Pi_{i,j}$) par intersection géométrique des blocs de P_i et de Q_j
2. Tri des blocs de redistribution dans un ordre canonique
3. Génération du message $\hat{m}_{i,j}$ par extraction des éléments associés aux blocs de redistribution

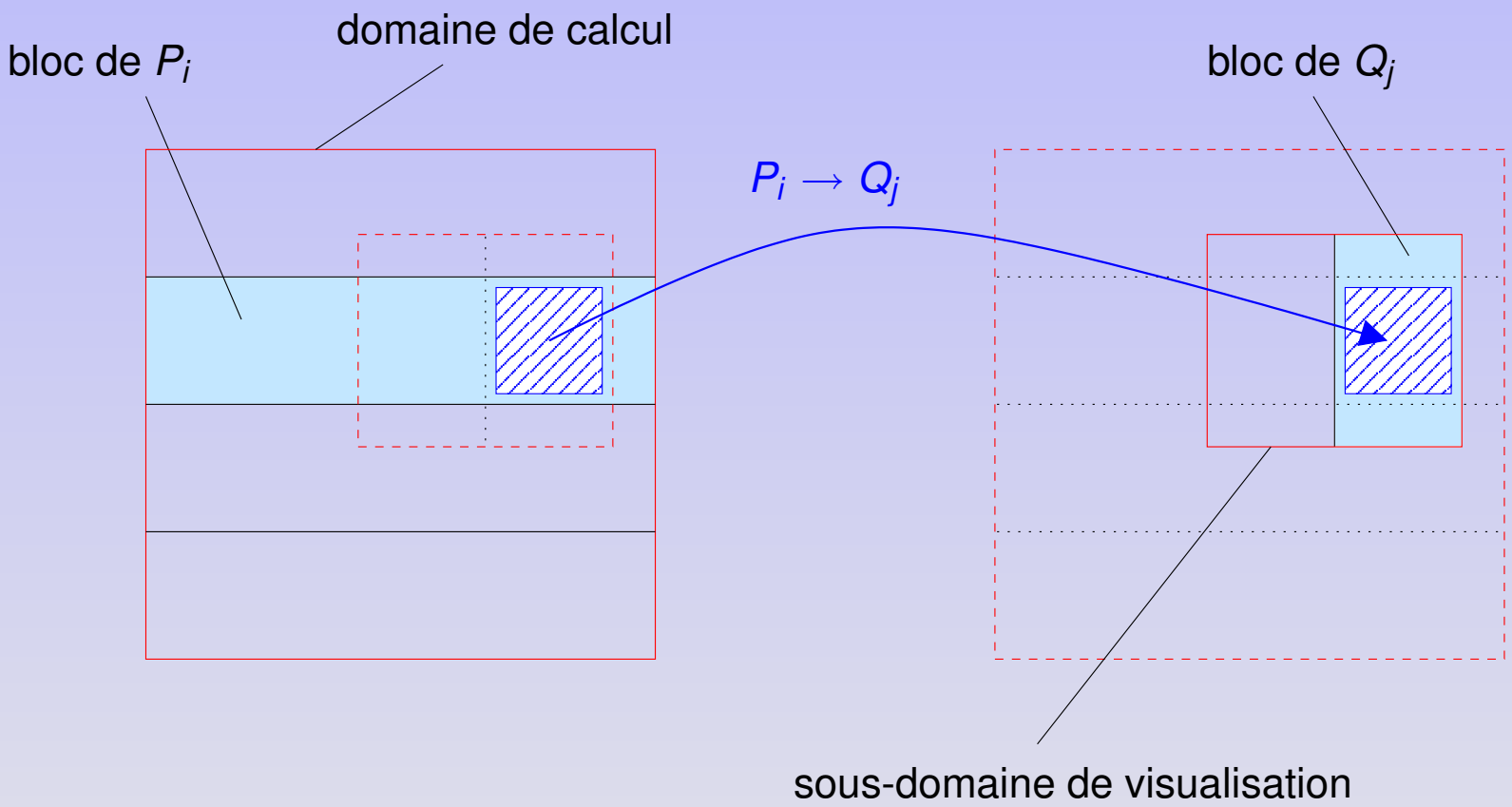


extraction dans le cas des boîtes d'atomes

Exemples

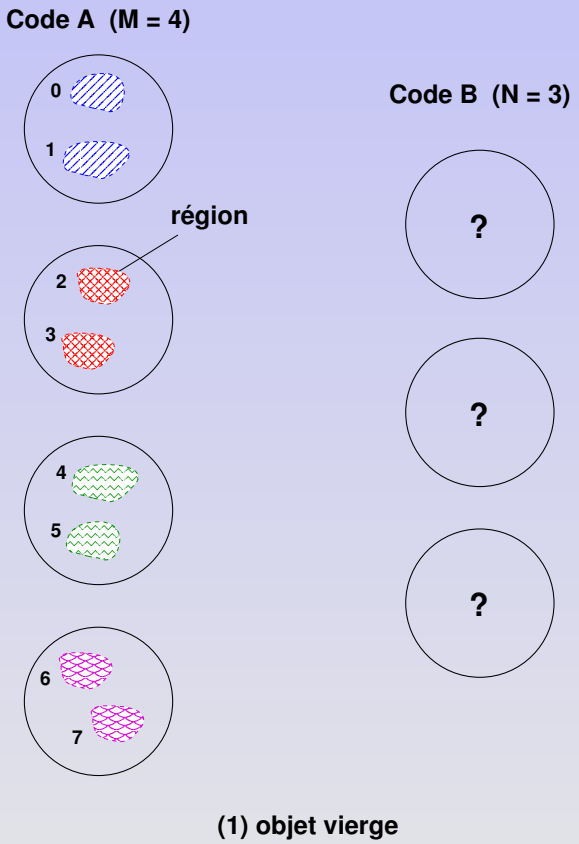

domaine
bloc de P_i
blocs de Q_j
$P_i \rightarrow Q_j$
blocs de redistribution $\Pi_{i,j}$
buffer

Exemples


bloc de P_i
domaine de calcul
$P_i \rightarrow Q_j$
bloc de Q_j
sous-domaine de visualisation

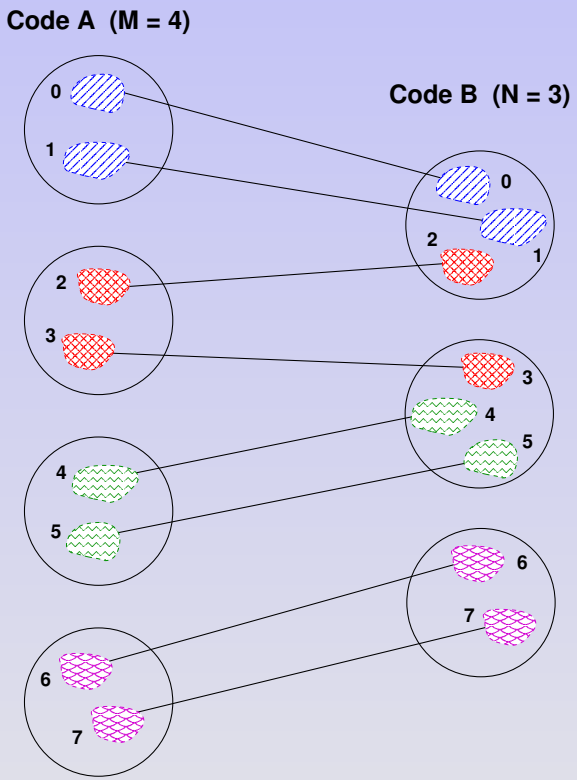
Approche placement sans découpage

- Application à tous les types d'objets complexes
- Calcul d'un placement des régions de A vers B (dépendant fonction de coût)
- Bien adapté au contexte du pilotage ($M > N$)



Approche placement sans découpage

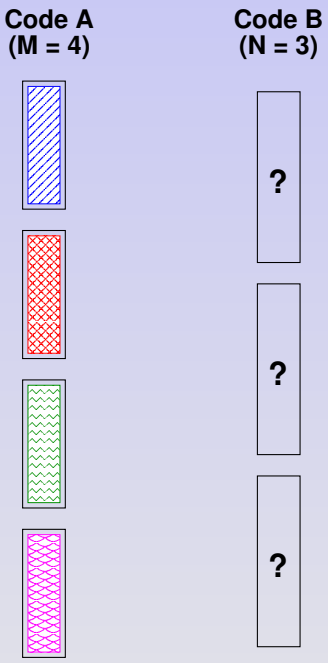
- Application à tous les types d'objets complexes
- Calcul d'un placement des régions de A vers B (dépendant fonction de coût)
- Bien adapté au contexte du pilotage ($M > N$)



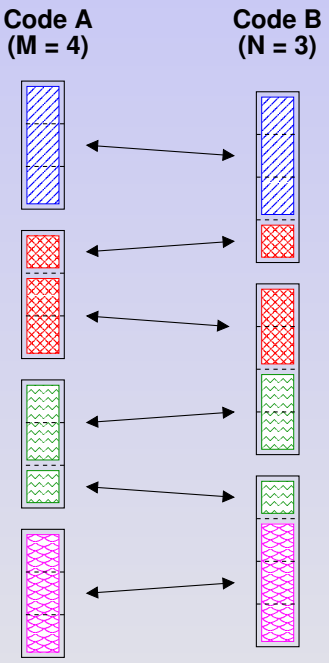
(2) résultat du placement

Approche placement avec découpage

- Application aux ensembles de particules et aux maillages non structurés
- Opération abstraite de découpage des régions
  $\longrightarrow$ triviale pour les ensembles de particules
  $\longrightarrow$ basée sur un partitionneur pour les maillages (ex. Metis, Scotch)
- Découpage en unités logiques affectées aux processus distants
  $\longrightarrow M + N - PGCD(M, N)$ messages si distribution initiale équilibrée



(1) objet vierge

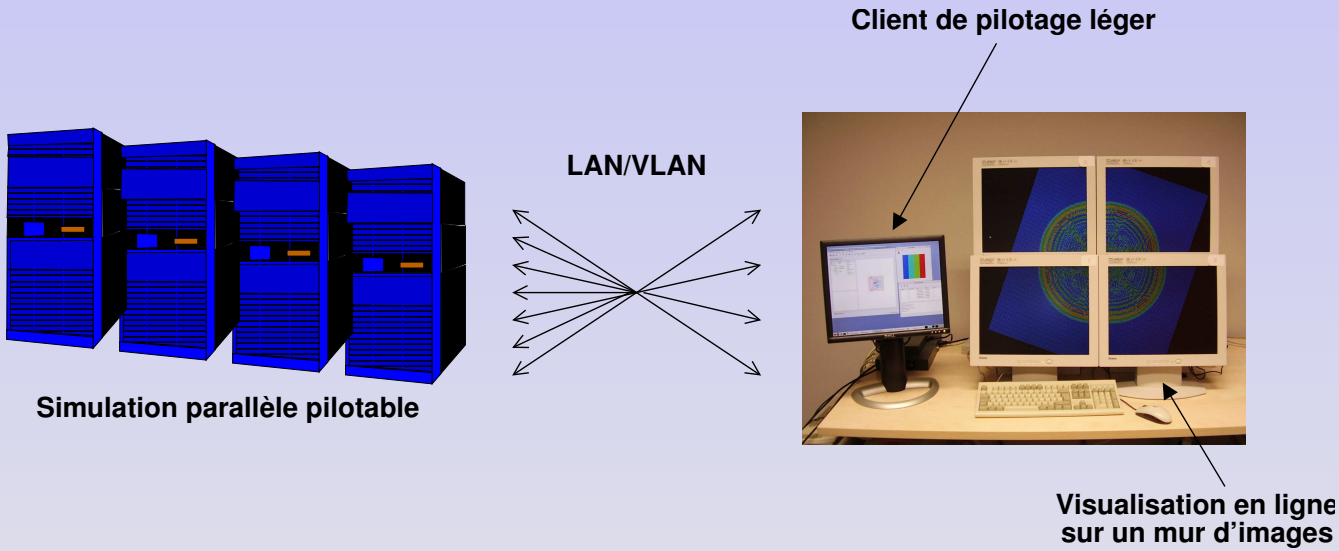


(2) résultat du placement

Plan

Principales fonctionnalités

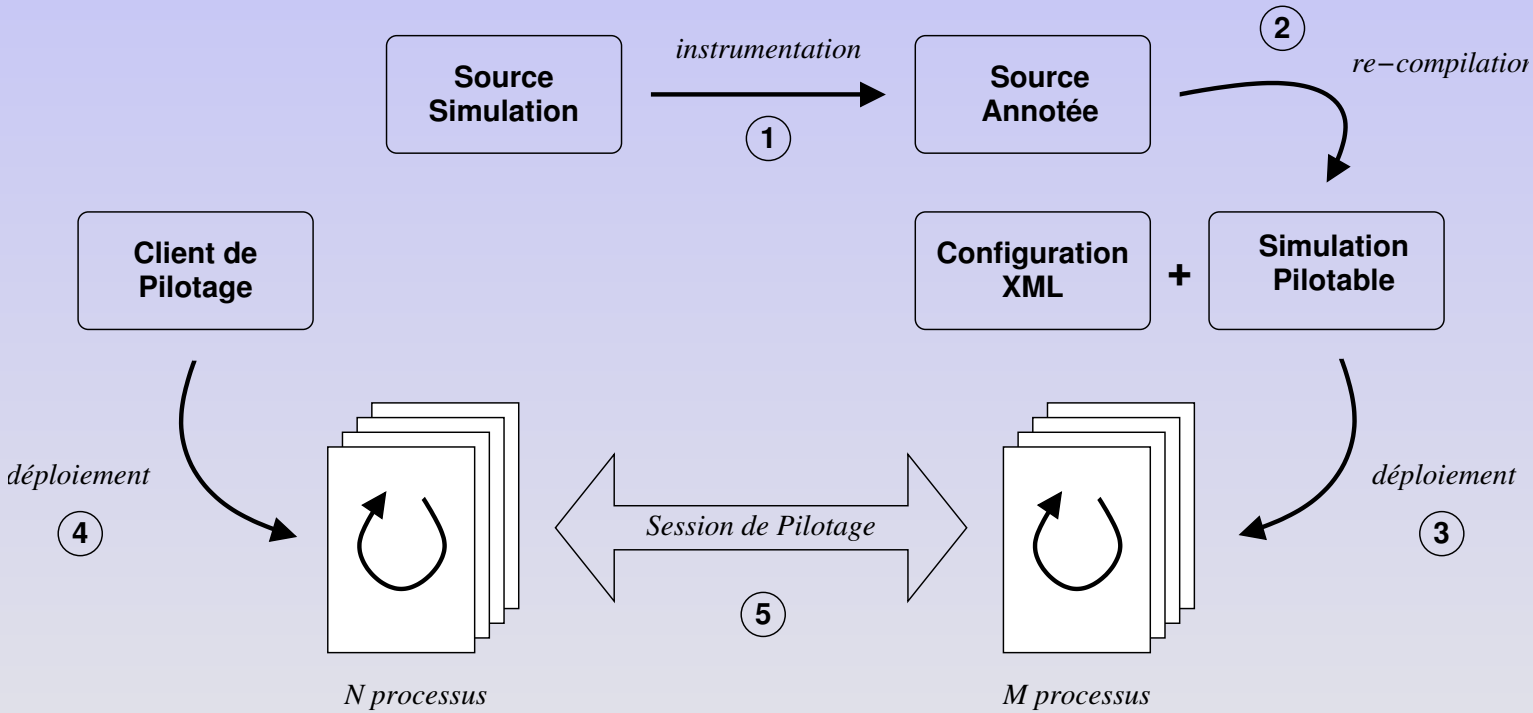
- Contrôle à distance du flot d'exécution
- Extraction de données pour la visualisation en ligne
- Modification des données à la volée
- Déclenchement d'actions (callback)



Environnement dynamique et distribué : connexion/déconnexion de multiples utilisateurs

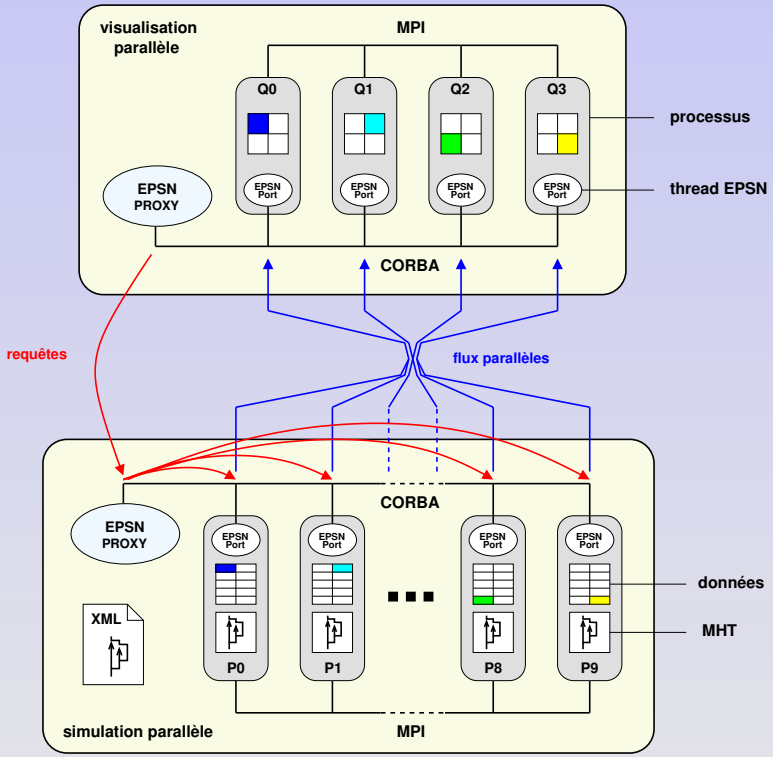
Processus d'intégration et de déploiement

- Annotation du code source + configuration des interactions via le fichier XML
- Re-compilation, déploiement de la simulation, connexions clientes
- Session de pilotage : requêtes asynchrones et concurrentes

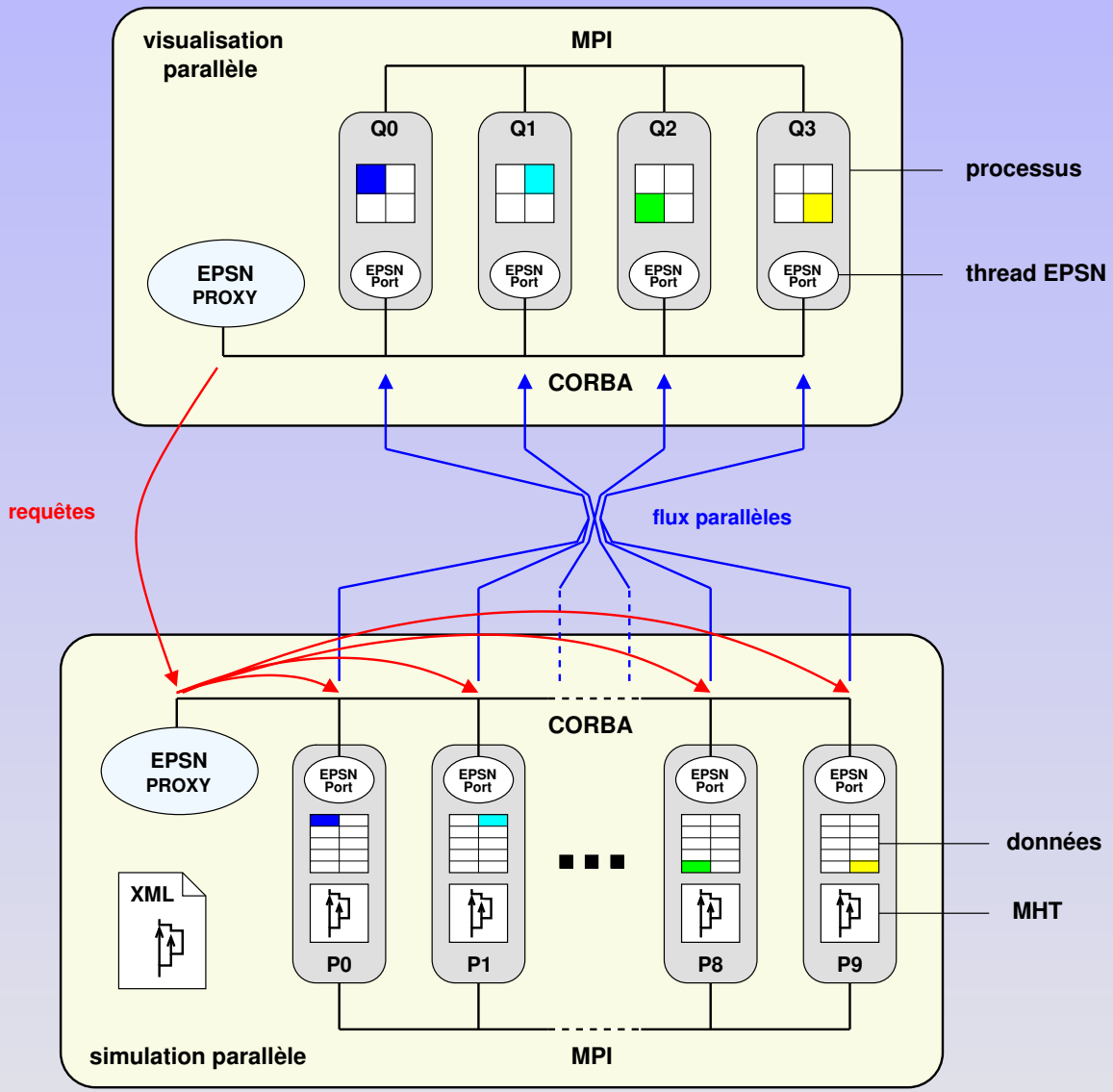


Couplage simulation parallèle et visualisation parallèle

- Approche composant parallèle (proxy + plusieurs ports)
- Gestion centralisée des requêtes, transfert direct des données
- Système de communication basé sur CORBA
- Threads EPSN dédiés aux traitements des interactions

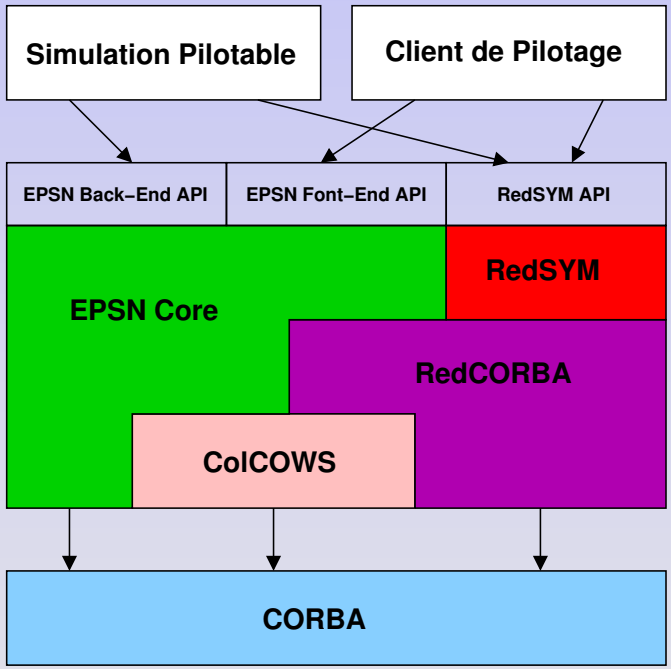


Architecture d'EPSN

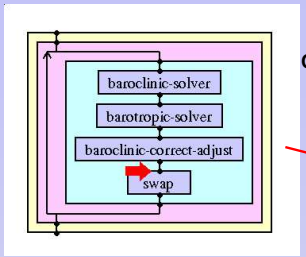


Vue en couche

- ColCOWS : couche de connexion CORBA
- RedSYM : couche de redistribution symbolique
- RedCORBA : couche de transfert basée sur CORBA
- EPSN Core : gestionnaire de requêtes, coordination, *etc.*



Interface utilisateur générique pour le pilotage de simulation EPSN



Visualisation du MHT

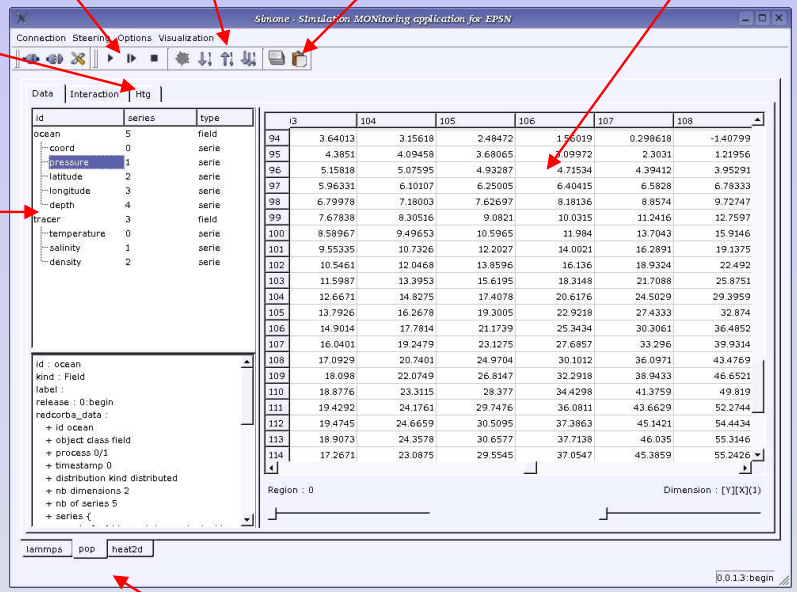
Accès aux Données (get/put/getp)

Contrôle (play/step/stop)

Séries de Données

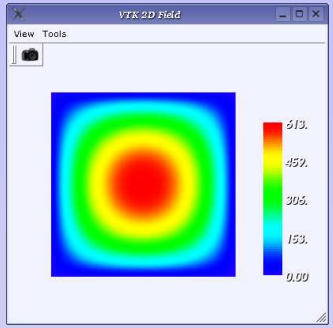
Benchmarks

Feuille de Données

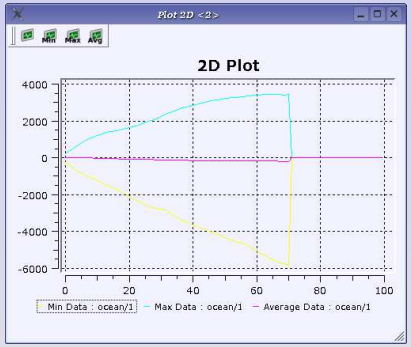


Simulations Connectées

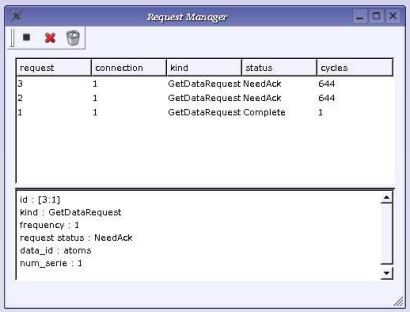
Date Courante



Plug–in de Visualisation VTK



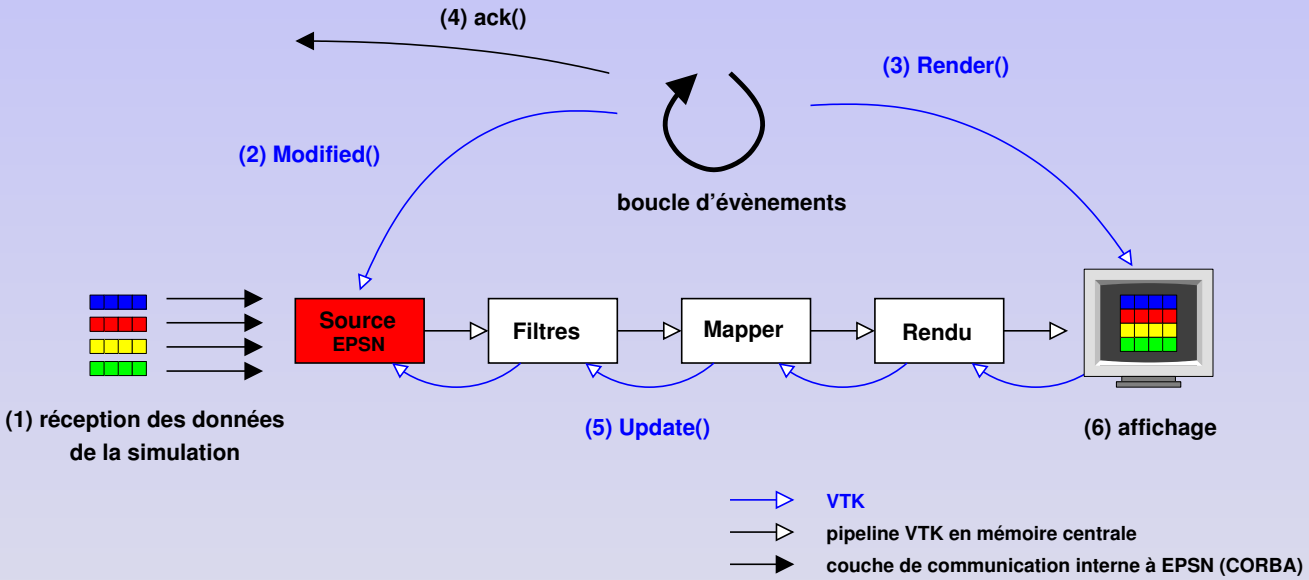
Plug–in de Visualisation QWT



Gestionnaire de Requêtes

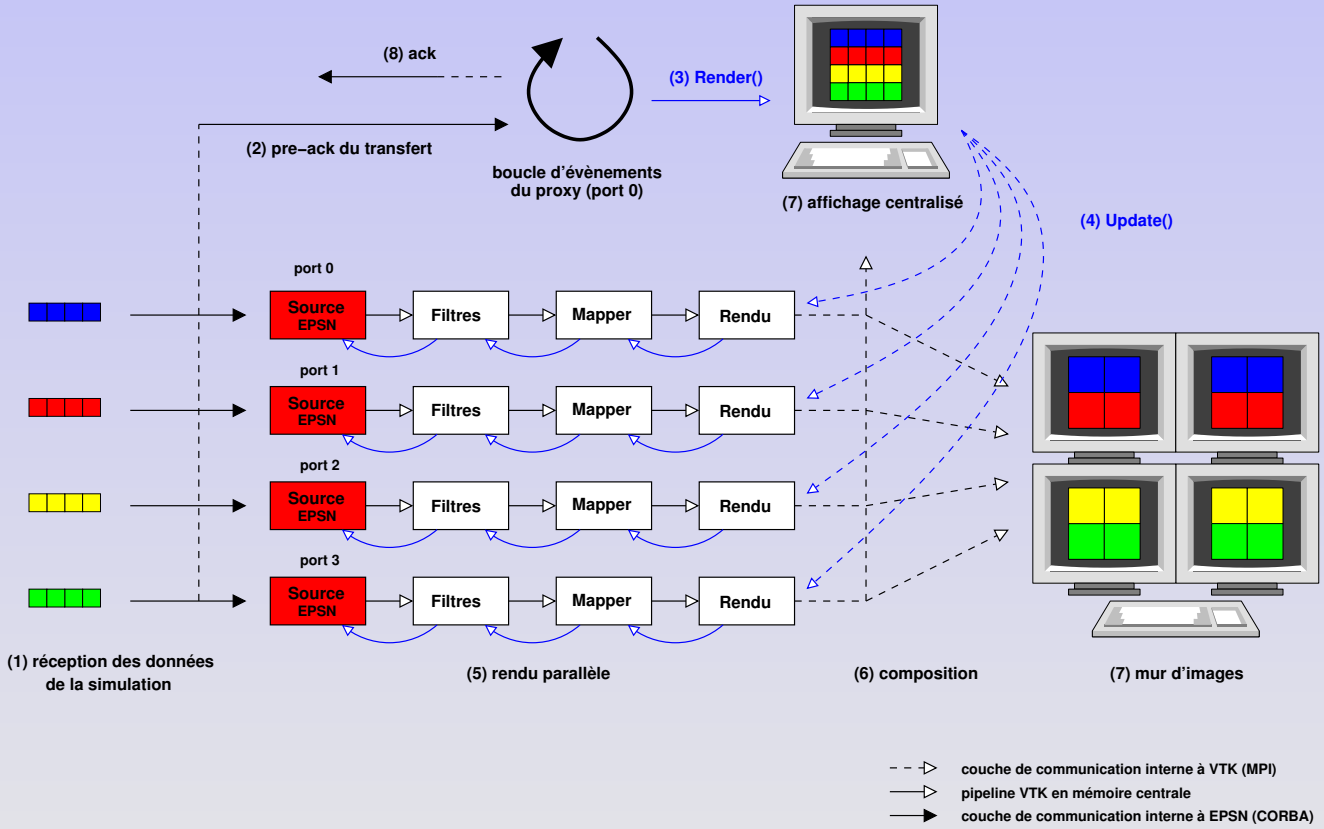
Visualisation en ligne avec VTK (The Visualization Toolkit)

- Traduction automatique des objets complexes RedSYM en source VTK
- Intégration transparente dans le pipeline de visualisation de VTK



Visualisation en ligne avec VTK (The Visualization Toolkit)

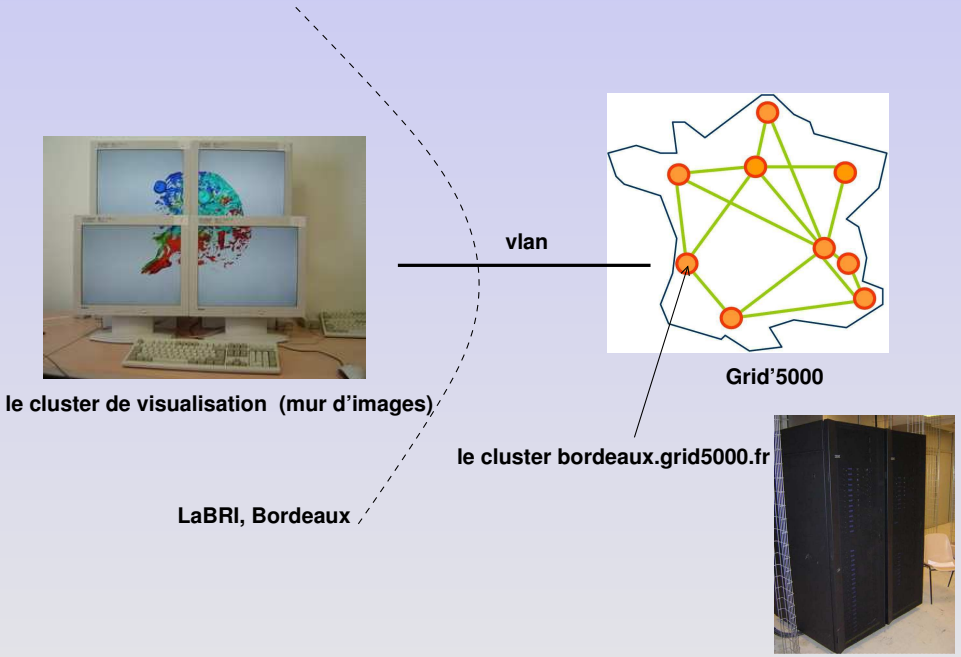
- Visualisation parallèle et rendu centralisé sur un écran
  → composition des rendus partiels en comparant le Z-buffer (*sort-last*)
- Visualisation parallèle et rendu parallèle sur un mur d'images
  → VTK + ICE-T (Moreland *et al.*)



Plan

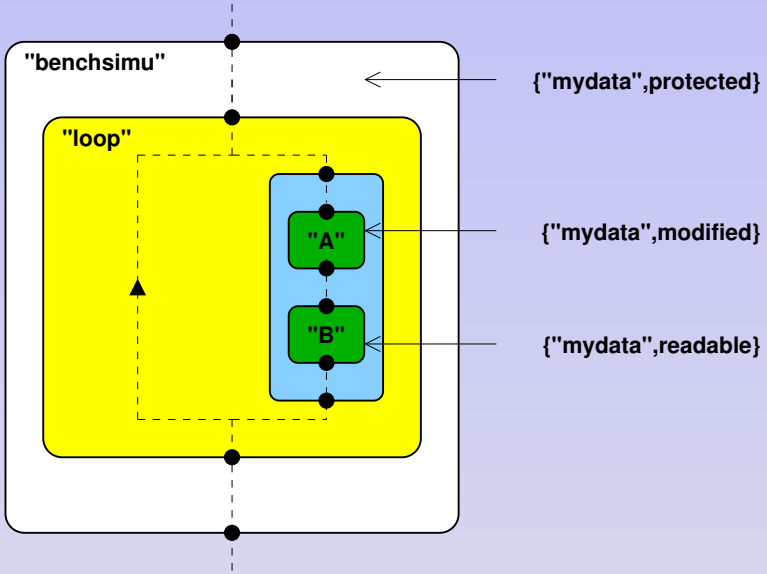
Conditions expérimentales

- Cluster de calcul Grid'5000 Bordeaux
  → 50 bi-opteron 2.2 GHz/2 Go
- Cluster de visualisation
  → 4 bi-xeon 2.8 GHz/2 Go, Nvidia GeForce 4 (AGP 8X, 128 Mo)
- Réseaux
  → giga-Ethernet intra-cluster (LAN) et inter-clusters (VLAN)
- Logiciels
  → LAM/MPI 7 : débit max. 112 Mo/s., latence 103 μs.
  → OmniORB 4 : débit max. 110 Mo/s. (avec copie 97.5 Mo/s.), latence 107 μs.



Mise en œuvre du recouvrement

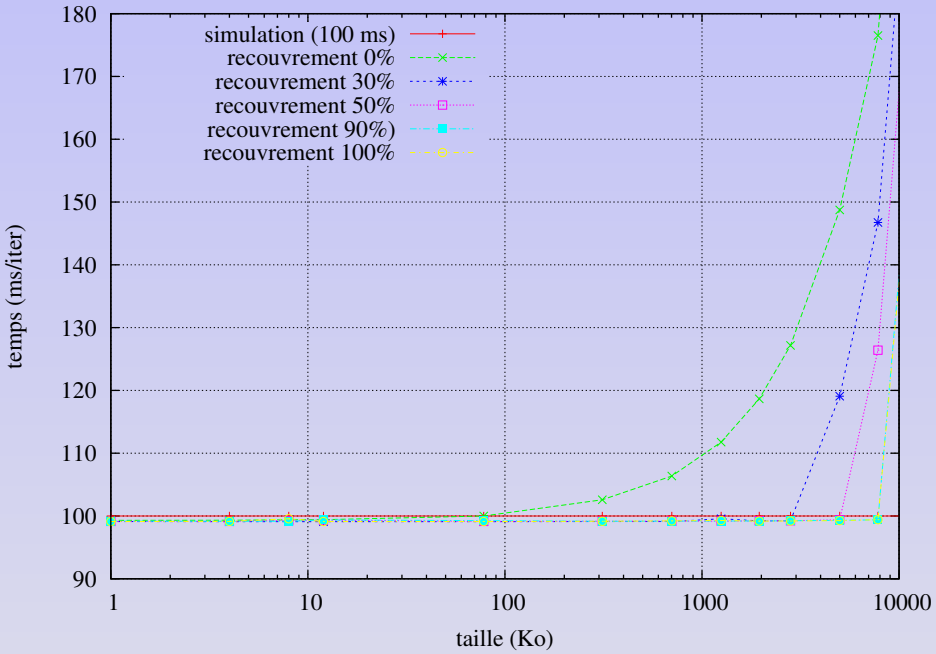
Simulation test avec deux tâches A et B. Variation de la plage de lecture (A) par rapport durée totale d'une itération (A+B). 1 transfert/iter.



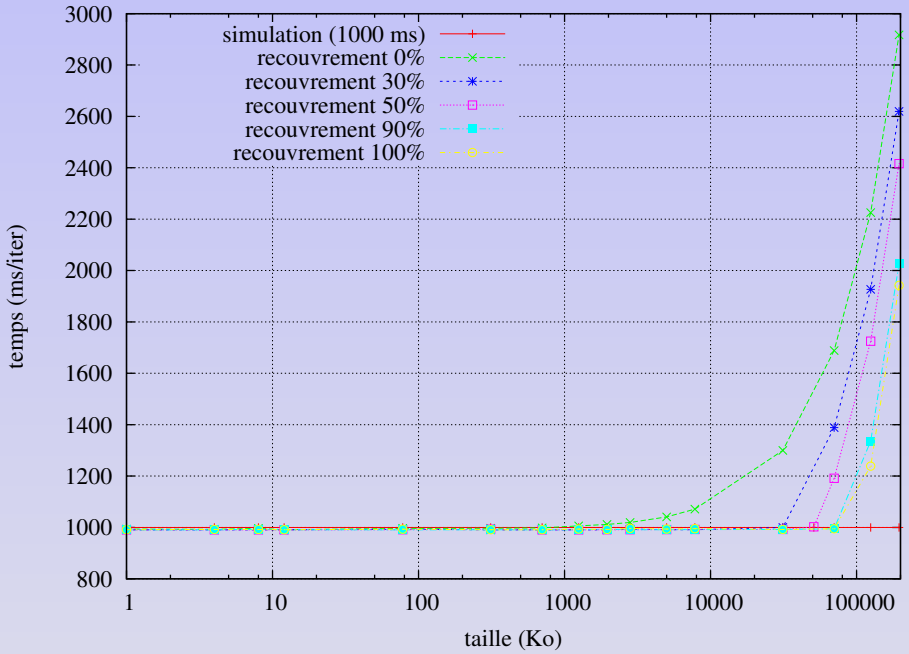
MHT de la simulation test.

Mise en œuvre du recouvrement

Simulation test avec deux tâches A et B. Variation de la plage de lecture (A) par rapport durée totale d'une itération (A+B). 1 transfert/iter.



Simulation rapide (100 ms/iter.).

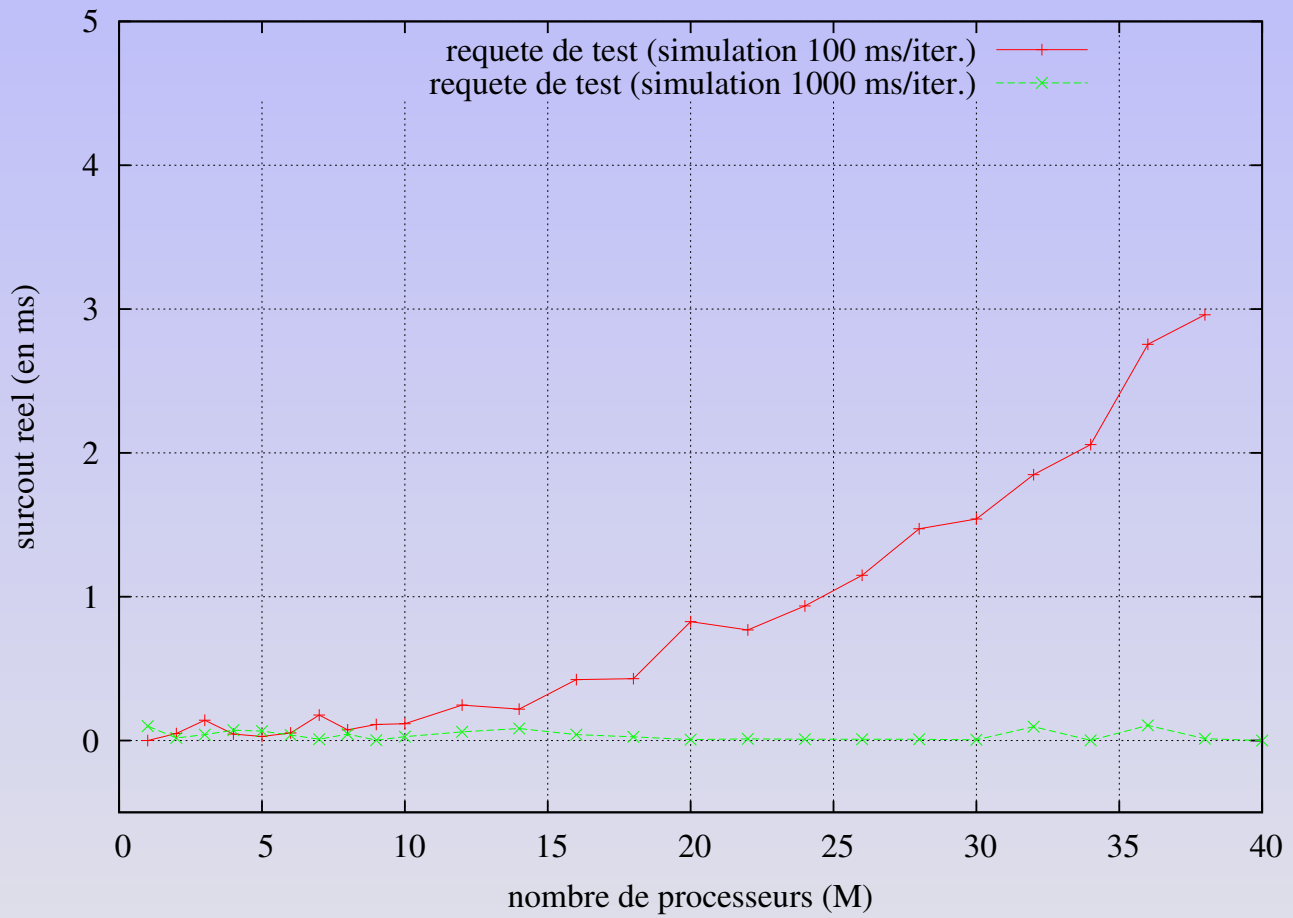


Simulation lente (1000 ms/iter.).

Bonne capacité d'EPSN à recouvrir les transferts

Évaluation de la phase de coordination

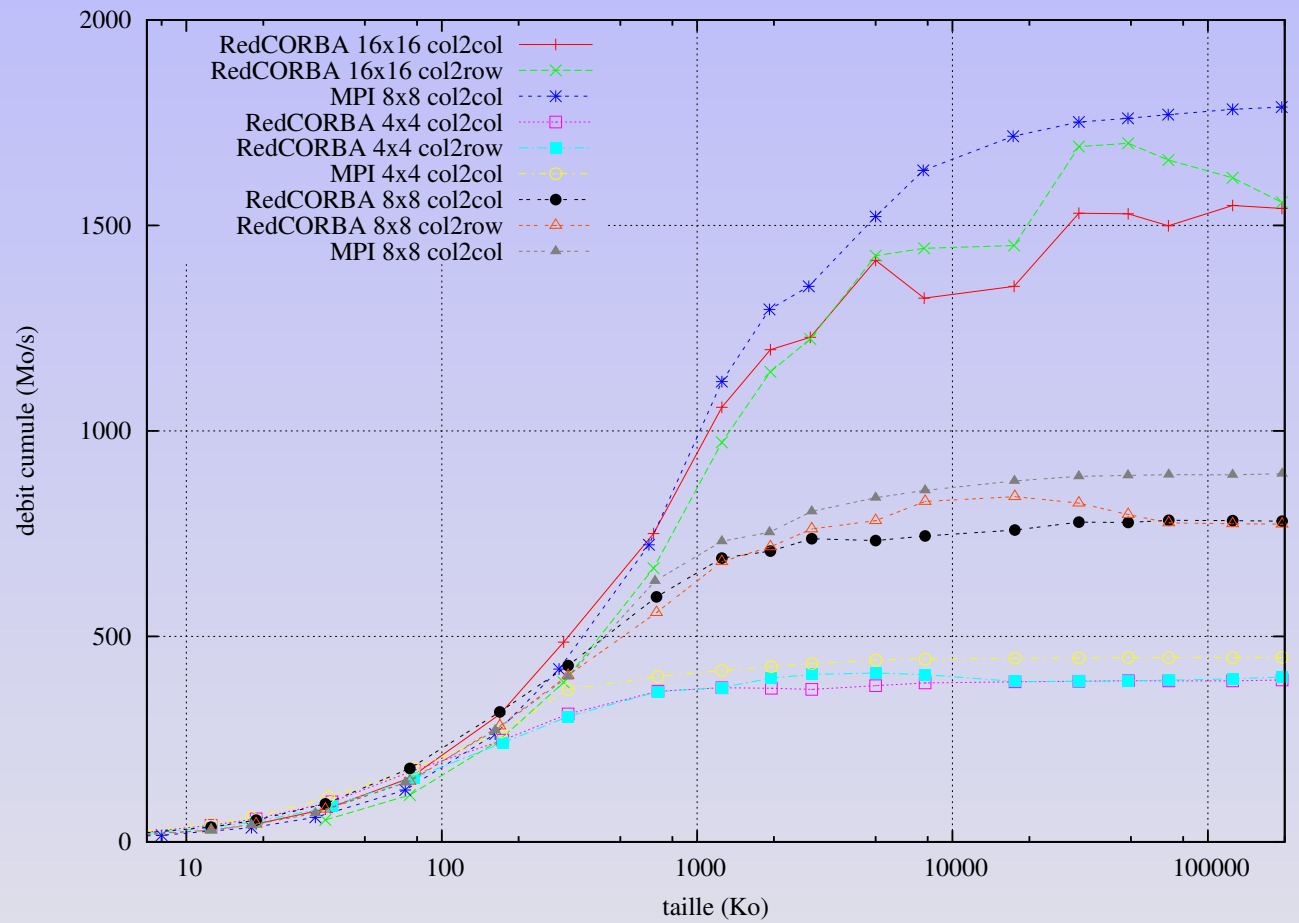
Surcoût réel pour une simulation rapide (100 ms/iter.) et lente (1000 ms/iter.).



Phase de coordination recouverte partiellement voire totalement

Redistribution des grilles structurées

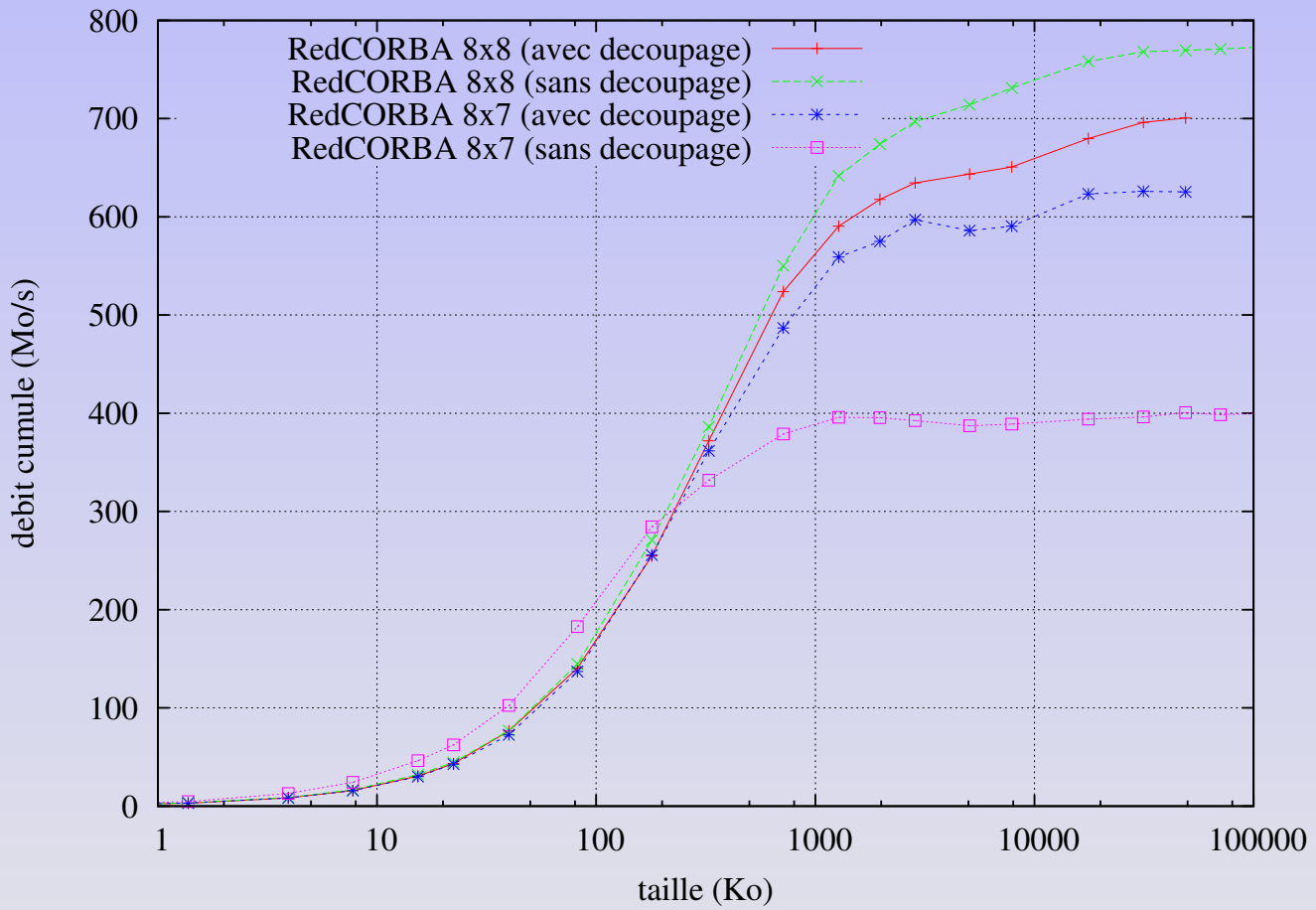
Cas de distributions *col2col* et *col2row* sur une grille rectangulaire 2D.



Bonne agrégation de la bande passante

Redistribution des maillages non structurés

Grille 2D stockée en non structuré (quad.). Distribution par colonne côté émetteur. Stratégie de découpage avec Metis. Série de données aux nœuds.



Stratégie avec découpage intéressante pour des cas irréguliers

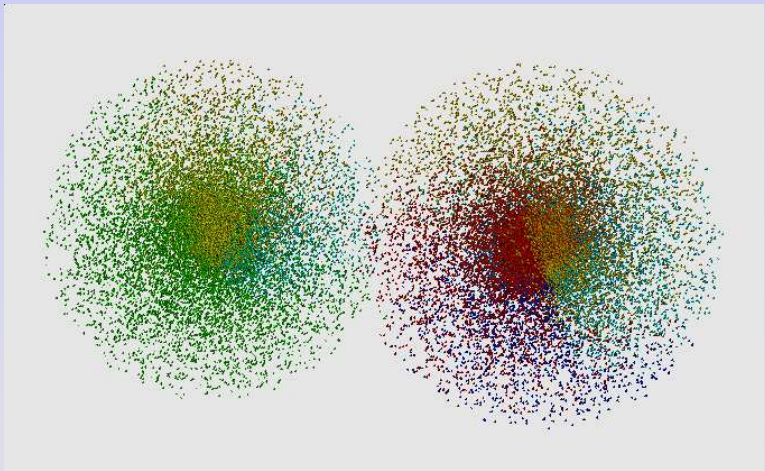
Plan

Gadget2

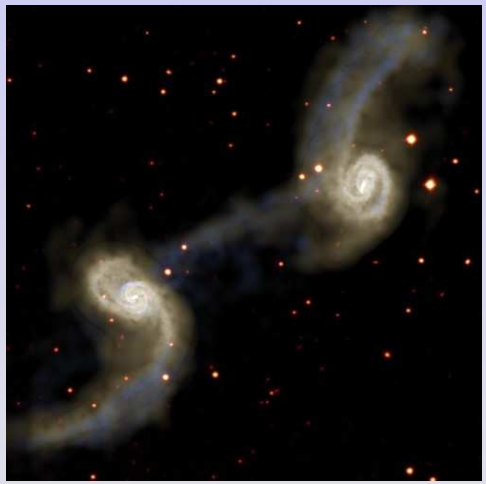
Simulation parallèle en astrophysique (Springel *et al.*)

- Galaxie modélisée par un fluide (Smooth Particle Hydrodynamics)
- Calcul des forces gravitationnelles (problème à N-corps) sur la base d'un *octree*
- Migration des particules très fréquente (dynamicité)
- Code C++ SPMD avec MPI

Cas test : collision de deux galaxies



Distribution sur 16 processeurs.



Collision de deux galaxies.

Résultats

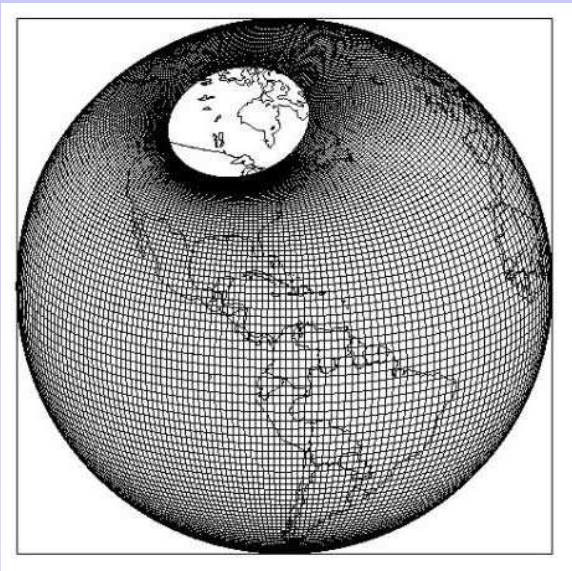
- 60000 particules (série des positions 3D, 1400 Ko)
- Simulation distribuée sur 16 processeurs
- Visualisation parallèle sur 4 processeurs (mur d'images 2×2)
- Visualisation des particules : glyph vs sprite
- Approche placement sans découpage : 4 régions par processus de visualisation
- Période d'envoi $p = 1$ ou 10

simulation à vide	258 ms/iter.	–
getp (p=1)	342 ms/iter.	+30%
getp (p=10)	279 ms/iter.	+8%
visu. séq. en ligne (p=1, glyph)	8250 ms/iter.	+3000%
visu. séq. en ligne (p=10, glyph)	1441 ms/iter.	+800%
visu. paral. en ligne ICE-T (p=1, glyph)	2370 ms/iter.	+450%
visu. paral. en ligne ICE-T (p=10, glyph)	379 ms/iter.	+46%
visu. paral. en ligne ICE-T (p=1, sprite)	370 ms/iter.	+43%
visu. paral. en ligne ICE-T (p=10, sprite)	280 ms/iter.	+8%

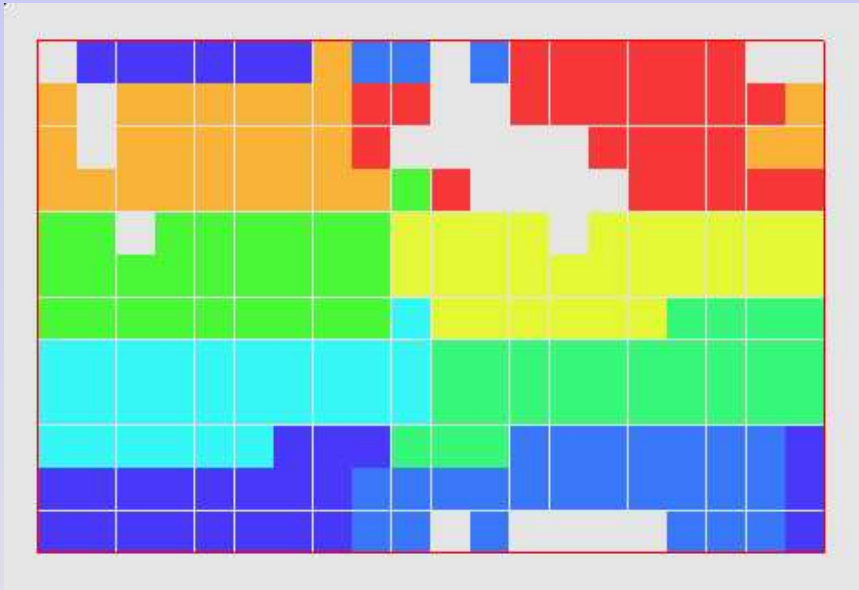
POP (Parallel Ocean Program)

Etude circulation océanique (Los Alamos, climate.lanl.gov/Models/POP)

- Dynamique des fluides sur une sphère en rotation
- Grille orthogonale 2D (760×468) $\rightarrow$ coordonnées sphériques
- Décomposition spatiale $\rightarrow$ distribution creuse par bloc
- Code Fortran 90, SPMD/MPI, tâche swap dans le MHT

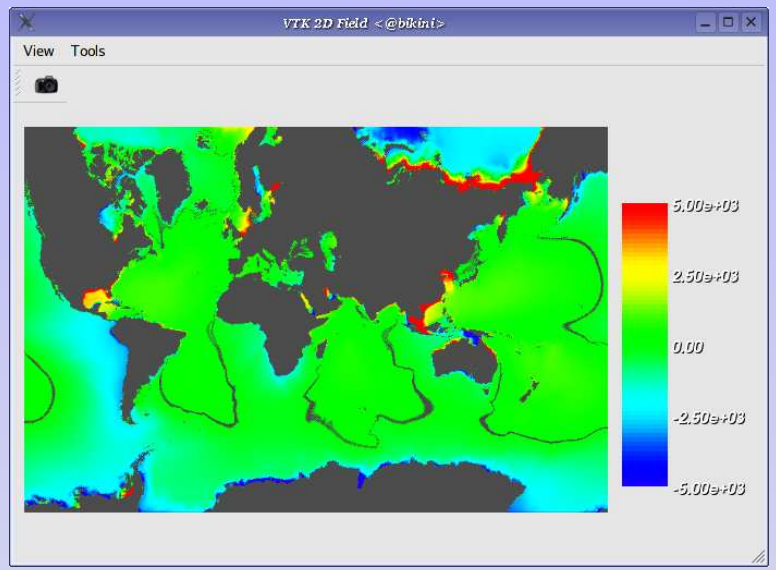


Grille irrégulière utilisée dans POP

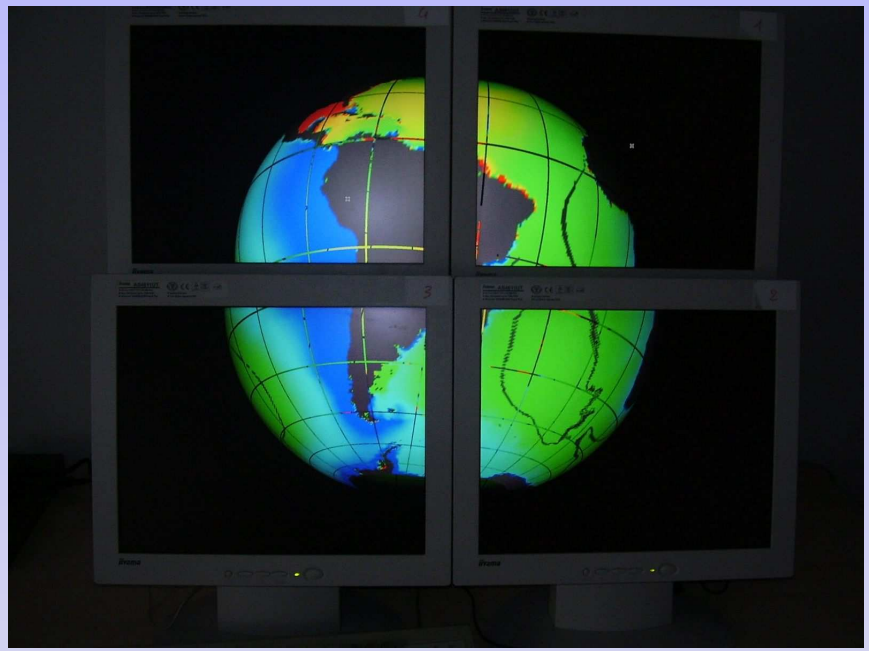


Distribution par bloc « creuse » sur 8 processeurs.

POP (Parallel Ocean Program)



Pression à la surface des océans (continent = gris)



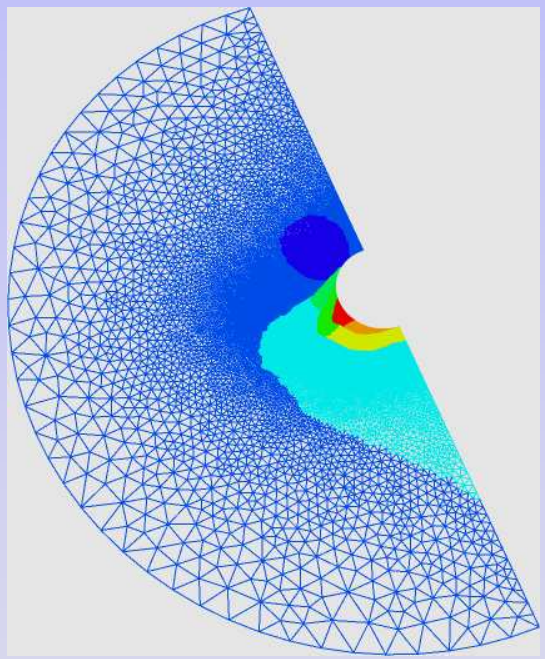
Visualisation en ligne sur un mur d'images 2 $\times$ 2.

Résultats préliminaires

- 12 s/iter. sur 8 processeurs, transfert pression (7897 Ko) 27 ms
- Reconstruction sphérique 3D : en séquentiel 20 s, contre 0.4 s en parallèle (ICE-T)

Simulation parallèle en mécanique des fluides (ScAlApplix)

- Mécanique des fluides : équation de Navier/Stokes 2D & 3D compressible
- Méthode explicite de type « volume finis »
- Cas test : écoulement hypersonique de l'air autour du cockpit d'une navette pénétrant dans l'atmosphère
- Maillage non structuré (16760 triangles)
- Code Fortran 90, SPMD/MPI



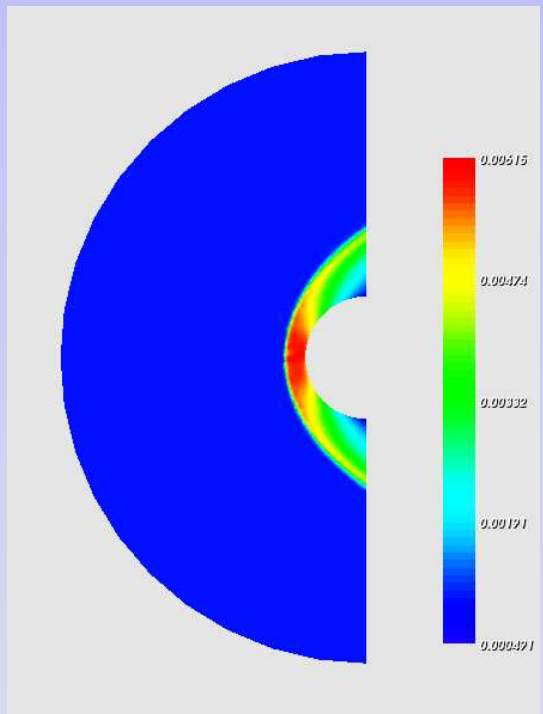
Distribution sur 8 processeurs.

Résultats préliminaires

- 2.35 s/iter. sur 8 processeurs
- Transfert densité (86 Ko) entièrement recouvert
- Visualisation en ligne séquentielle +12%/iter.
- Visualisation en ligne parallèle en +2.5%/iter.

Simulation parallèle en mécanique des fluides (ScAlApplix)

- Mécanique des fluides : équation de Navier/Stokes 2D & 3D compressible
- Méthode explicite de type « volume finis »
- Cas test : écoulement hypersonique de l'air autour du cockpit d'une navette pénétrant dans l'atmosphère
- Maillage non structuré (16760 triangles)
- Code Fortran 90, SPMD/MPI



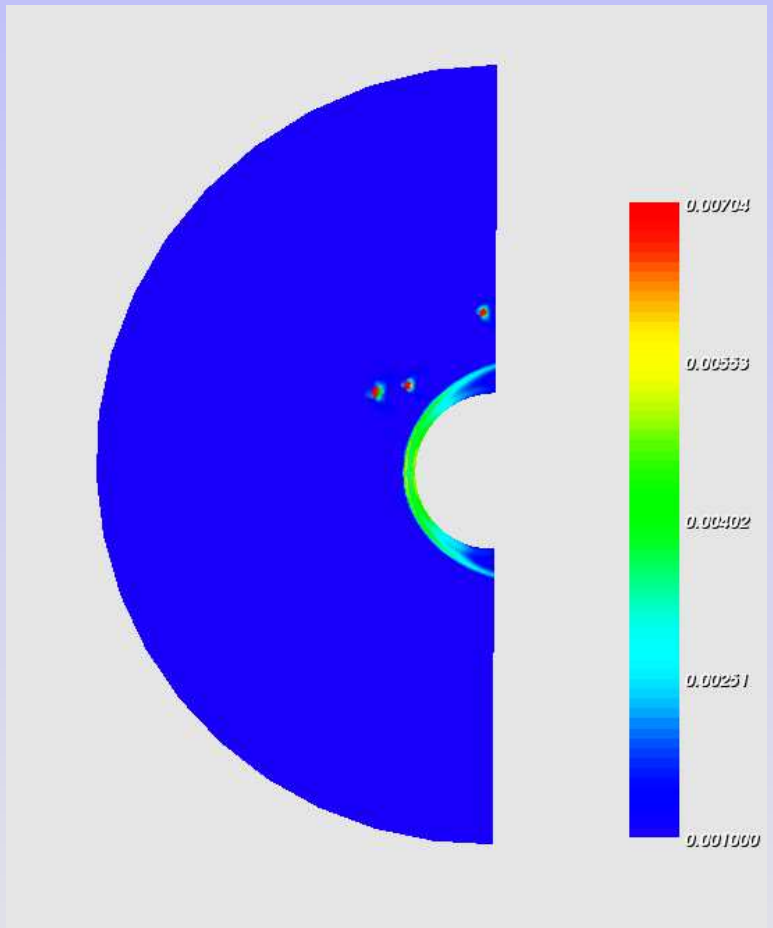
Visualisation de la densité de l'air (VTK).

Résultats préliminaires

- 2.35 s/iter. sur 8 processeurs
- Transfert densité (86 Ko) entièrement recouvert
- Visualisation en ligne séquentielle +12%/iter.
- Visualisation en ligne parallèle en +2.5%/iter.

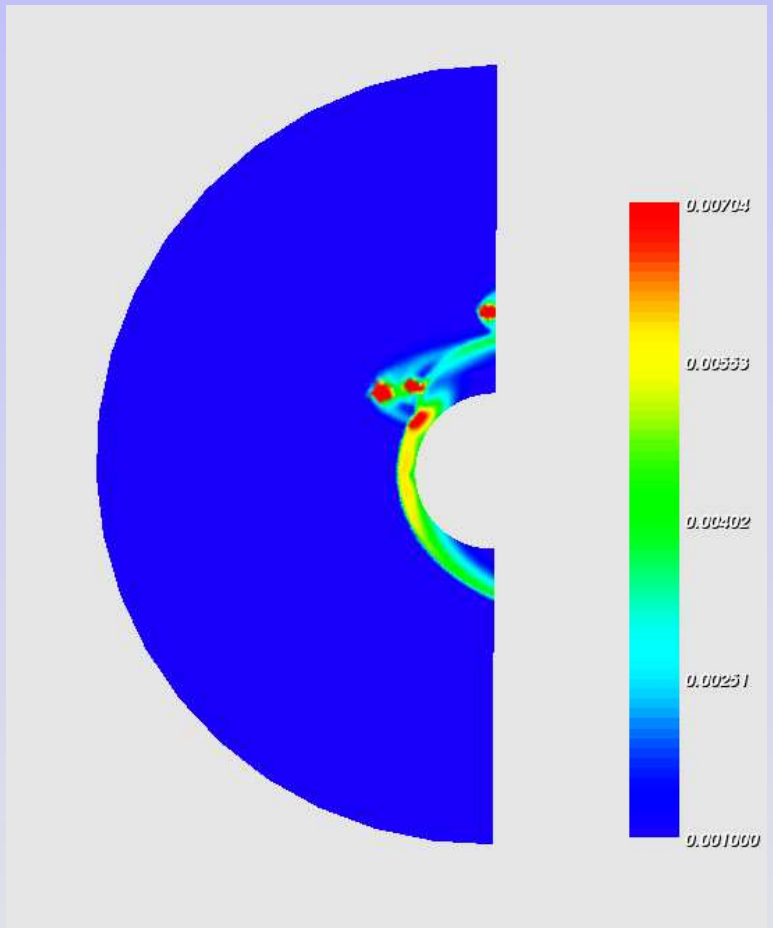
Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



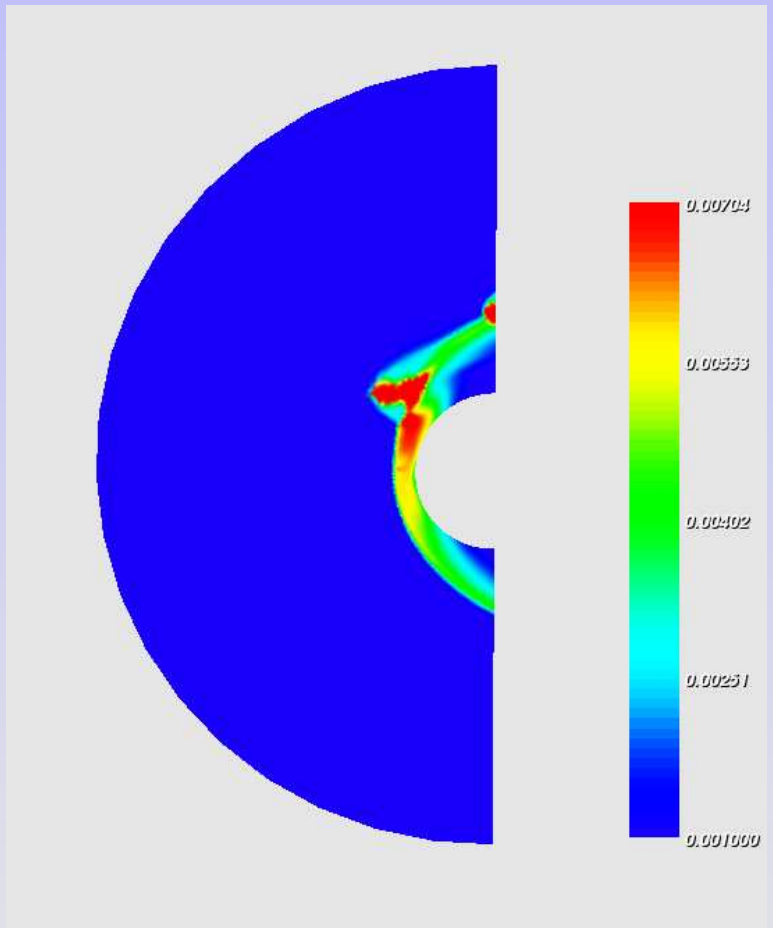
Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



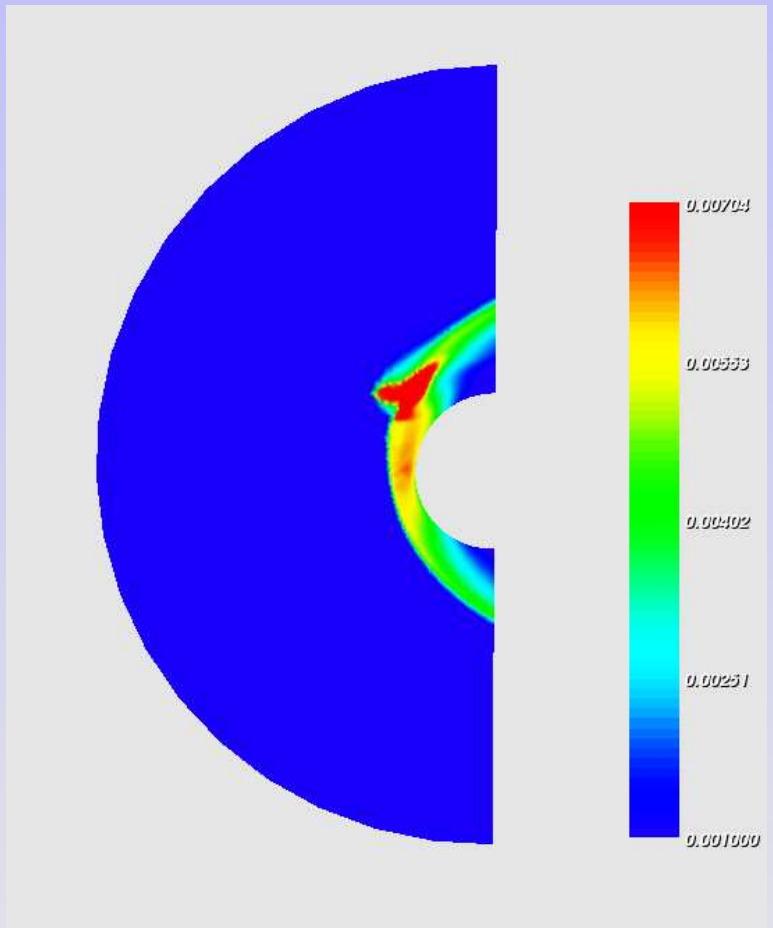
Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



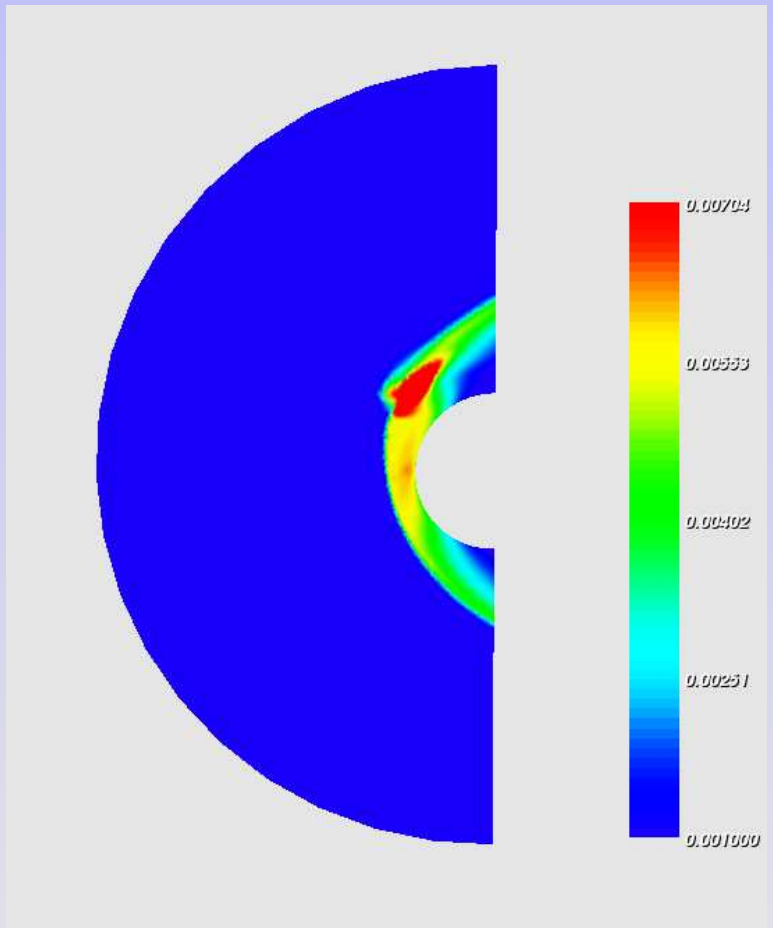
Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



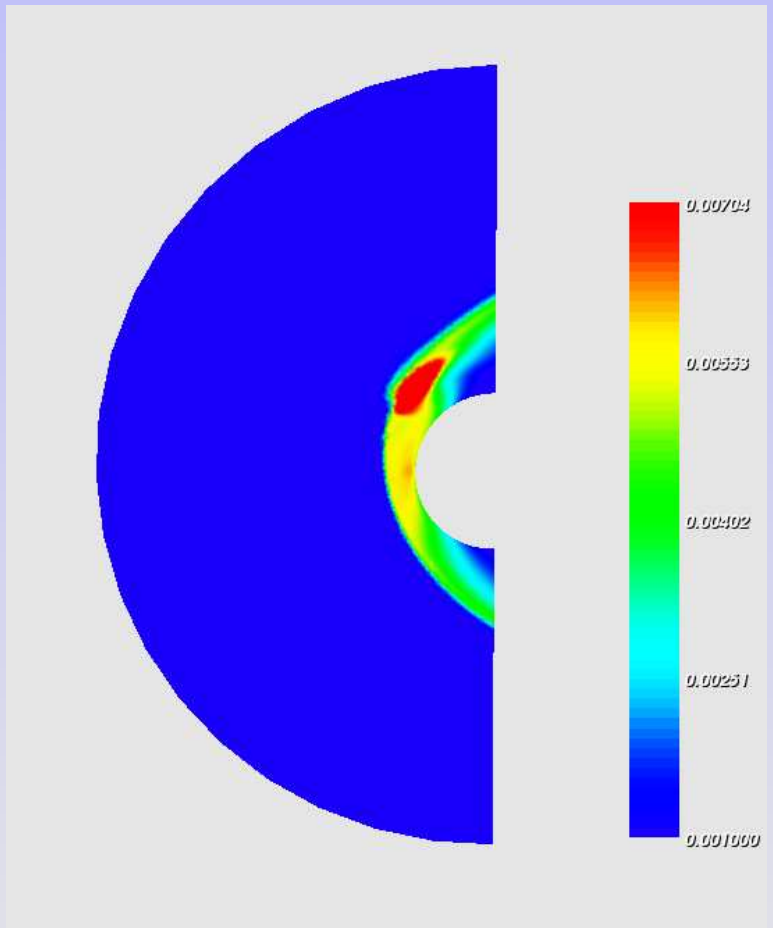
Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



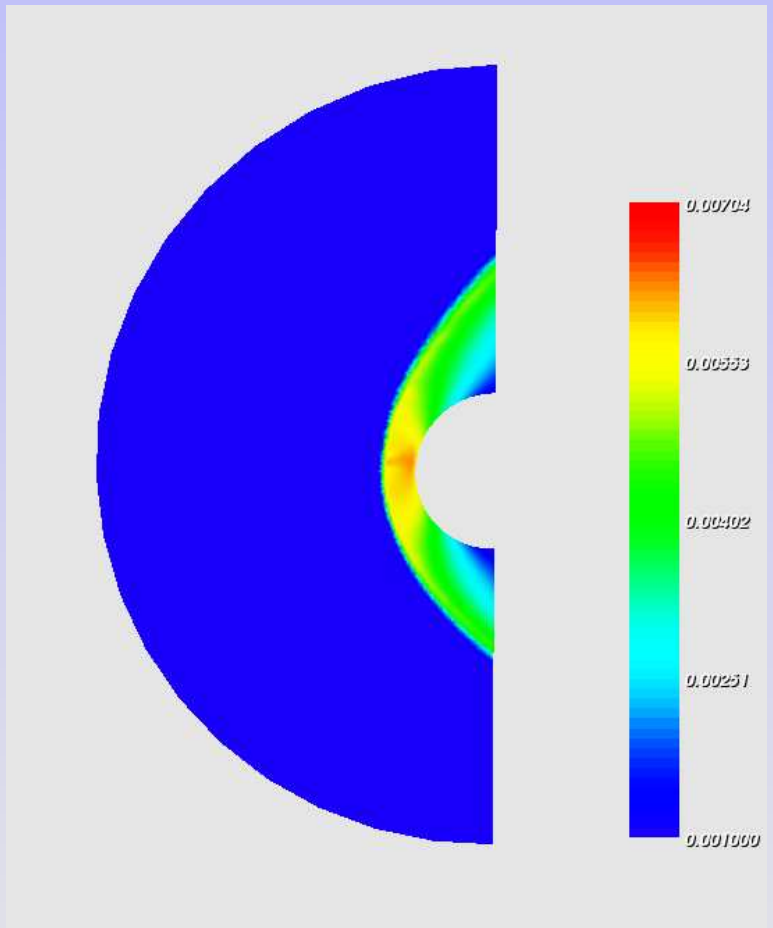
Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



Perturbation à la volée de la densité du fluide avec Simone

→ un exemple concret d'interaction pratiqué « à la main » par les scientifiques



Conclusion

Environnement de pilotage pour des simulations parallèles intégrant des capacités de visualisation parallèle

- Modélisation des simulations SPMD sur la base d'un MHT
  → introduction d'un système de datation pour garantir la cohérence des interactions (accès aux données)
- Proposition d'un modèle de redistribution pour des objets complexes
  → messages symboliques, approches spatiale et placement de la redistribution
- Conception et réalisation de la plate-forme EPSN
  → plate-forme s'appuyant sur ces deux modèles
  → couche de communication basée sur CORBA
  → visualisation parallèle avec VTK & ICE-T
- Mesures des performances et validation sur des « vrais » codes
  → recouvrement des transferts, coordination efficace, agrégation de la bande-passante, bénéfice de la visualisation parallèle

Perspectives

Perspectives à court terme

- Amélioration des performances des communication
  → mise en place d'une autre couche de transfert : PadicoTM, MPI-2, ...
  → ajouter des schémas de filtrage et de compression des données
  → ordonnancement des communications (KBPS, Wagner & Jeannot)
- Vers la réalité virtuelle
  → interaction dans un environnement immersif : VTK+VRPN, CAT
- Test de passage à l'échelle avec Grid'5000 !

Perspectives à plus long terme

- Vers des objets plus complexes
  → objets hiérarchiques (octrees, grilles multi-niveaux, arbres de partitionnement)
  → prise en charge de la dynamicité (fenêtre de visualisation, AMR)
  → redistribution entre des grilles et des maillages non identiques
- Vers le pilotage des simulations parallèles avec couplage de codes
  → simulations multi-physiques, multi-échelles
- Développement d'un serveur de données relai
  → rejouer et visualiser l'historique de la simulation hors-ligne